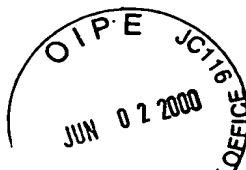


09/521,098



日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1998年 7月14日

出 願 番 号

Application Number:

平成10年特許願第198737号

出 願 人

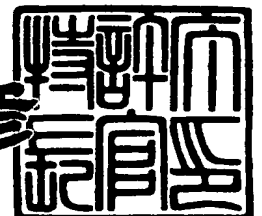
Applicant (s):

ソニー株式会社

2000年 4月21日

特許庁長官  
Commissioner,  
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3028433

【書類名】 特許願

【整理番号】 9800667001

【提出日】 平成10年 7月14日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 5/00

【発明の名称】 データ伝達制御方法

【請求項の数】 10

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
内

    【氏名】 村田 賢一

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
内

    【氏名】 北里 直久

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
内

    【氏名】 斎藤 潤也

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
内

    【氏名】 片山 靖

【特許出願人】

    【識別番号】 000002185

    【氏名又は名称】 ソニー株式会社

    【代表者】 出井 伸之

【代理人】

    【識別番号】 100086841

【弁理士】

【氏名又は名称】 脇 篤夫

【代理人】

【識別番号】 100102635

【弁理士】

【氏名又は名称】 浅見 保男

【手数料の表示】

【予納台帳番号】 014650

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9710074

【包括委任状番号】 9711279

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ伝達制御方法

【特許請求の範囲】

【請求項 1】 所定方式のシナリオ記述により関連付けされるオブジェクトを含んでデータ伝送単位を形成して、更に、或る一定期間にわたり転送すべき 1 以上のデータ伝送単位を含んで循環データ単位を形成し、この循環データ単位を繰り返し転送するデータ転送方式に対応し、

上記循環データ単位に含まれるオブジェクトについて更新が行われたことを、上記循環データ単位を受信するサーバ側からこの循環データ単位を使用するクライアントに対して通知できるようにしたオブジェクト更新通知処理、

を実行するように構成されていることを特徴とするデータ伝達制御方法。

【請求項 2】 上記循環データ単位には、上記オブジェクトの更新に応じてその内容が更新される上記データ伝送単位に関する制御情報が含まれるものとされ、

上記オブジェクト更新通知処理は、

所定のインターフェ이스の規格のもとで、上記制御情報の受取りがあったことを意味する制御情報更新イベントを設定したうえで、

少なくとも、

クライアントが、制御情報更新イベントの受取りを宣言するためのサブスクライブイベントをサーバに伝達する第 1 の処理と、

クライアントがイベントの発生の通知を要求するイベント通知要求をサーバに伝達する第 2 の処理と、

上記イベント通知要求に対するサーバの応答として、更新された制御情報を受信したときには、制御情報更新イベントが発生したことをクライアントに通知する第 3 の処理と、

を実行することにより実現されることを特徴とする請求項 1 に記載のデータ伝達制御方法。

【請求項 3】 上記第 3 の処理は、更新された制御情報により示されるデータ伝送単位の識別情報を付加して、制御情報受取りイベントが発生したことをク

クライアントに通知するようにされていることを特徴とする請求項2に記載のデータ伝達制御方法。

【請求項4】 上記第1の処理としてのサブスクライブイベントの伝達は、クライアントのプログラムの立ち上げ時、又は、上記循環データ単位自体の切り換えが行われたときに実行するように構成されていること特徴とする請求項2に記載のデータ伝達制御方法。

【請求項5】 上記第2の処理としてのイベント通知要求は、サーバからのイベント通知要求に対する応答メッセージが得られた後において、直ちに実行するように構成されていることを特徴とする請求項2に記載のデータ伝達制御方法。

【請求項6】 上記循環データ単位には、属するオブジェクトの更新に応じてその内容が更新されるデータ伝送単位に関する制御情報が含まれるものとされ、

上記オブジェクト更新通知処理は、

少なくとも、

クライアントが、制御情報更新イベントの受取りを宣言するためのサブスクライブイベントをサーバに伝達する第1の処理と、

クライアントにて関心のあるオブジェクトのオブジェクトIDをサーバ側に通知し、サーバ側では通知されたオブジェクトが属するデータ伝送単位のデータ伝送単位IDをクライアントに返送する第2の処理と、

クライアントにおいて、上記第2の処理にてサーバ側に通知したオブジェクトIDと、上記第2の処理によりサーバ側から獲得したデータ伝送単位IDとの対応を示すテーブル情報を生成する第3の処理と、

クライアントが、イベントの発生の通知を要求するイベント通知要求をサーバに伝達する第4の処理と、

サーバにおける上記イベント通知要求に対する応答処理として、サーバにおいて更新された制御情報を受信したときには、この更新された制御情報により示されるデータ伝送単位IDを付加して、制御情報更新イベントが発生したことをクライアントに通知する第5の処理と、

クライアントにおいて、上記第5の処理により得られたデータ伝送単位IDと一致したテーブル情報のデータ伝送単位IDを検索し、この検索されたデータ伝送単位のIDに対応付けされたオブジェクトIDが示すオブジェクトについて更新が行われたものと特定する第6の処理と、

を実行するように構成されていることを特徴とする請求項1に記載のデータ伝達制御方法。

【請求項7】 上記第1の処理としてのサブスクライブイベントの伝達は、クライアントのプログラムの立ち上げ時、又は、上記循環データ単位自体の切り換えが行われたときに実行するように構成されていること特徴とする請求項6に記載のデータ伝達制御方法。

【請求項8】 上記第4の処理としてのイベント通知要求は、サーバからのイベント通知要求に対する応答メッセージが得られた後において、直ちに実行されるように構成されていることを特徴とする請求項6に記載のデータ伝達制御方法。

【請求項9】 上記循環データ単位には、属するオブジェクトの更新に応じてその内容が更新されるデータ伝送単位に関する制御情報が含まれるものとされ

上記オブジェクト更新通知処理は、

クライアントが、データ伝送単位更新イベントの受取りを宣言するためのサブスクライブイベントを関心のあるオブジェクトのオブジェクトIDと共にサーバに伝達し、サーバは、これに応答して固有のデータ伝送単位更新イベントIDを設定してクライアントに伝達する第1の処理と、

クライアントにおいて、上記第1の処理にてサーバ側に伝達したオブジェクトIDと、上記第1の処理によりサーバ側から獲得したデータ伝送単位更新イベントIDとの対応を示すテーブル情報を生成する第2の処理と、

クライアントが、データ伝送単位更新イベントの発生の通知を要求するイベント通知要求をサーバに伝達する第3の処理と、

サーバにおける上記イベント通知要求に対する応答として、更新された制御情報を受信したときには、この更新された制御情報により示されるデータ伝送単位

に含まれるオブジェクトのオブジェクトIDに対応して設定した上記データ伝送単位更新イベントIDを付加して、データ伝送単位更新イベントが発生したことをクライアントに通知する第4の処理と、

クライアントにおいて、上記第4の処理により得たデータ伝送単位更新イベントIDと一致する上記テーブル情報のデータ伝送単位更新イベントIDに対応するオブジェクトIDを識別し、この識別したオブジェクトIDが示すオブジェクトを更新されたオブジェクトとして特定する第5の処理と、

を実行するように構成されていることを特徴とする請求項1に記載のデータ伝達制御方法。

【請求項10】 上記第2の処理としてのイベント通知要求は、サーバからのイベント通知要求に対する応答メッセージが得られた後において、直ちに実行するように構成されていることを特徴とする請求項9に記載のデータ伝達制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばデジタル衛星放送などにおいてデータサービスを受信するシステムに適用して好適なデータ伝達制御方法に関するものである。

【0002】

【従来の技術】

近年、デジタル衛星放送の普及が進んでいる。デジタル衛星放送は、例えば既存のアナログ放送と比較してノイズやフェージングに強く、高品質の信号を伝送することが可能である。また、周波数利用効率が向上され、多チャンネル化も図ることが可能になる。具体的には、デジタル衛星放送であれば1つの衛星で数百チャンネルを確保することも可能である。このようなデジタル衛星放送では、スポーツ、映画、音楽、ニュースなどの専門チャンネルが多数用意されており、これらの専門チャンネルでは、それぞれの専門のコンテンツに応じたプログラムが放送されている。

【0003】

そして、上記のようなデジタル衛星放送システムを利用して、ユーザが楽曲等の音声データをダウンロードできるようにしたり、いわゆるテレビショッピングとして、例えばユーザが放送画面を見ながら何らかの商品についての購買契約を結べるようにしたりすることが提案されている。つまりは、デジタル衛星放送システムとして、通常の放送内容と並行したデータサービス放送を行うものである。

【0004】

一例として、楽曲データのダウンロードであれば、放送側においては、放送番組と並行して、楽曲データを多重化して放送するようにする。また、この楽曲データのダウンロードに際しては、GUI (Graphical User Interface)画面（即ちダウンロード用の操作画面である）を表示させることでインタラクティブな操作をユーザに行わせるようにされるが、このGUI画面出力のためのデータも多重化して放送するようにされる。

そして、受信装置を所有しているユーザ側では、所望のチャンネルを選局している状態で、受信装置に対する所定の操作によって楽曲データをダウンロードするためのGUI画面を表示出力させるようにする。そして、この表示された操作画面に対してユーザが操作を行うことで、例えば受信装置に接続したデジタルオーディオ機器に対してデータを供給し、これが録音されるようにするものである。

【0005】

ところで、上記のような楽曲データをダウンロードするためのGUI画面としては、例えばGUI画面を形成するパーツ的な画像データ、テキストデータなどの情報に加え、更には所定操作に応じた音声出力のための音声データなどの単位データ（ファイル）をそれぞれオブジェクトとして扱い、このオブジェクトの出力態様を所定方式によるシナリオ記述によって規定することによって、上記操作画面についての所要の表示形態及び音声等の出力態様を実現するように構成することが考えられる。

なお、ここでは、上記GUI画面のようにして、記述情報によって規定される



ことで、或る目的に従った機能を実現する表示画面（ここでは音声等の出力も含む）のことを「シーン」というものとする。また、「オブジェクト」とは、記述情報に基づいてその出力態様が規定される画像、音声、テキスト等の単位情報を示しており、伝送時においては、ここでは記述情報自体のデータファイルも「オブジェクト」の1つとして扱われるものとする。

#### 【0006】

上記シーン表示及びシーン表示上での音声出力等を実現するためのオブジェクトは、例えば所定の伝送方式に従ってエンコードされて送信される。

受信装置側では上記伝送方式に従ってデータを受信すると共に、この受信データについてデコード処理を施して、例えば表示に必要なシーンに必要とされるオブジェクトごとの纏まりとしてのデータを得て、これをシーンとして出力するようにされる。

#### 【0007】

##### 【発明が解決しようとする課題】

ここで、受信装置を所有するユーザの使用環境を考慮すれば、上記所定の伝送方式に則った上で、受信装置にて受信したデータサービス用のデータの処理については出来るだけ効率的な処理が行われるようにして、例えば表示出力されるべきシーン内容の更新なども、できるだけ軽い処理で迅速に対応できるようにすることが好ましい。

#### 【0008】

##### 【課題を解決するための手段】

そこで、本発明は上記した課題を考慮して、所定方式のシナリオ記述により関連付けされるオブジェクトを含んでデータ伝送単位を形成して、更に、或る一定期間にわたり転送すべき1以上のデータ伝送単位を含んで循環データ単位を形成し、この循環データ単位を繰り返し転送するデータ転送方式に対応したデータ伝達制御方法として、上記循環データ単位に含まれるオブジェクトについて更新が行われたことを、循環データ単位を受信するサーバ側からこの循環データ単位を使用するクライアントに対して通知できるようにしたオブジェクト更新通知処理を実行するように構成することとした。

【0009】

上記構成によれば、少なくとも、循環データ単位を形成するデータについて、何らかのオブジェクトが更新されたことをクライアントが知ることが可能となり、クライアントでは、これに基づいた所要の対応処理を実行することが可能になる。

【0010】

【発明の実施の形態】

以降、本発明の実施の形態について説明する。

本発明が適用されるシステムとしては、デジタル衛星放送を利用して番組を放送すると共に、受信装置側ではこの番組に関連した楽曲データ（音声データ）等の情報をダウンロードできるようにしたシステムを例に挙げることにする。

【0011】

なお、以降の説明は次の順序で行うこととする。

1. デジタル衛星放送システム

1-1. 全体構成

1-2. GUI画面に対する操作

1-3. 地上局

1-4. 送信フォーマット

1-5. IRD

2. 本発明に至った背景

3. 本実施の形態のオブジェクト更新通知制御

3-1. U-U APIの一般的処理

3-2. 第1例

3-3. 第2例

3-4. 第3例

【0012】

1. デジタル衛星放送システムの構成

## 1-1. 全体構成

図1は、本実施の形態としてのデジタル衛星放送システムの全体構成を示すものである。この図に示すように、デジタル衛星放送の地上局1には、テレビ番組素材サーバ6からのテレビ番組放送のための素材と、楽曲素材サーバ7からの楽曲データの素材と、音声付加情報サーバ8からの音声付加情報と、GUIデータサーバからのGUIデータとが送られる。

## 【0013】

テレビ番組素材サーバ6は、通常の放送番組の素材を提供するサーバである。このテレビ番組素材サーバから送られてくる音楽放送の素材は、動画及び音声とされる。例えば、音楽放送番組であれば、上記テレビ番組素材サーバ6の動画及び音声の素材を利用して、例えば新曲のプロモーション用の動画及び音声が放送されたりすることになる。

## 【0014】

楽曲素材サーバ7は、オーディオチャンネルを使用して、オーディオ番組を提供するサーバである。このオーディオ番組の素材は音声のみとなる。この楽曲素材サーバ7は、複数のオーディオチャンネルのオーディオ番組の素材を地上局1に伝送する。

各オーディオチャンネルの番組放送ではそれぞれ同一の楽曲が所定の単位時間繰り返して放送される。各オーディオチャンネルは、それぞれ独立しており、その利用方法としては各種考えられる。例えば、1つのオーディオチャンネルでは最新の日本のポップスの数曲を或る一定時間繰り返し放送し、他のオーディオチャンネルでは最新の外国のポップスの数曲を或る一定時間繰り返し放送するというようにされる。

## 【0015】

音声付加情報サーバ8は、楽曲素材サーバ7から出力される楽曲の時間情報等を提供するサーバである。

## 【0016】

GUIデータサーバ9は、ユーザが操作に用いるGUI画面を形成するための

「GUIデータ」を提供する。例えば後述するような楽曲のダウンロードに関するGUI画面であれば、配信される楽曲のリストページや各楽曲の情報ページを形成するための画像データ、テキストデータ、アルバムジャケットの静止画を形成するためのデータなどを提供する。更には、受信設備3側にていわゆるEPG(Electrical Program Guide)といわれる番組表表示を行うのに利用されるEPGデータもここから提供される。

なお、「GUIデータ」としては、例えばMHEG(Multimedia Hypermedia Information Coding Experts Group)方式が採用される。MHEGとは、マルチメディア情報、手順、操作などのそれぞれと、その組み合わせをオブジェクトとして捉え、それらのオブジェクトを符号化したうえで、タイトル(例えばGUI画面)として制作するためのシナリオ記述の国際標準とされる。また、本実施の形態ではMHEG-5を採用するものとする。

#### 【0017】

地上局1は上記テレビ番組素材サーバ6、楽曲素材サーバ7、音声付加情報サーバ8、及びGUIデータサーバ9から伝送された情報を多重化して送信する。

本実施の形態では、テレビ番組素材サーバ6から伝送されたビデオデータはMPEG(Moving Picture Experts Group)2方式により圧縮符号化され、オーディオデータはMPEG2オーディオ方式により圧縮符号化される。また、楽曲素材サーバ7から伝送されたオーディオデータは、オーディオチャンネルごとに対応して、例えばMPEG2オーディオ方式と、ATRAC(Adaptive Transform Acoustic Coding)方式と何れか一方の方式により圧縮符号化される。

また、これらのデータは多重化の際、キー情報サーバ10からのキー情報を利用して暗号化される。

なお、地上局1の内部構成例については後述する。

#### 【0018】

地上局1からの信号は衛星2を介して各家庭の受信設備3で受信される。衛星2には複数のトランスポンダが搭載されている。1つのトランスポンダは例えば30Mbpsの伝送能力を有している。各家庭の受信設備3としては、パラボラアンテナ11とIRD(Integrated Receiver Decoder)12と、ストレージデバ

イス 13 と、モニタ装置 14 とが用意される。

また、この場合には、IRD 12 に対して操作を行うためのリモートコントローラ 64 が示されている。

【0019】

パラボラアンテナ 11 で衛星 2 を介して放送されてきた信号が受信される。この受信信号がパラボラアンテナ 11 に取り付けられた LNB (Low Noise Block Down Converter) 15 で所定の周波数に変換され、IRD 12 に供給される。

【0020】

IRD 12 における概略的な動作としては、受信信号から所定のチャンネルの信号を選局し、その選局された信号から番組としてのビデオデータ及びオーディオデータの復調を行ってビデオ信号、オーディオ信号として出力する。また、IRD 12 では、番組としてのデータと共に多重化されて送信されてくる、GUI データに基づいて GUI 画面としての出力も行う。このような IRD 12 の出力は、例えばモニタ装置 14 に対して供給される。これにより、モニタ装置 14 では、IRD 12 により受信選局した番組の画像表示及び音声出力が行われ、また、後述するようなユーザの操作に従って GUI 画面を表示させることが可能となる。

【0021】

ストレージデバイス 13 は、IRD 12 によりダウンロードされたオーディオデータ（楽曲データ）を保存するためのものである。このストレージデバイス 13 の種類としては特に限定されるものではなく、MD (Mini Disc) レコーダ/プレーヤ、DAT レコーダ/プレーヤ、DVD レコーダ/プレーヤ等を用いることができる。また、ストレージデバイス 13 としてパーソナルコンピュータ装置を用い、ハードディスクのほか、CD-R 等をはじめとする記録が可能なメディアにオーディオデータを保存するようにすることも可能とされる。

【0022】

また、本実施の形態の受信設備 3 としては、図 2 に示すように、データ伝送規格として IEEE 1394 に対応したデータインターフェイスを備えた MD レコーダ/プレーヤ 13A を、図 1 に示すストレージデバイス 13 として使用するこ

とができるようになっている。

この図に示す IEEE 1394 対応の MD レコーダ/プレーヤ 13A は、IEEE 1394 バス 16 により IRD 12 と接続される。これによって、本実施の形態では、IRD 12 にて受信された、楽曲としてのオーディオデータ（ダウンロードデータ）を、ATRAC 方式により圧縮処理が施されたままの状態に直接取り込んで記録することができる。また、MD レコーダ/プレーヤ 13A と IRD 12 とを IEEE 1394 バス 16 により接続した場合には、上記オーディオデータの他、そのアルバムのジャケットデータ（静止画データ）及び歌詞などのテキストデータを記録することも可能とされている。

#### 【0023】

IRD 12 は、例えば電話回線 4 を介して課金サーバ 5 と通信可能とされている。IRD 12 には、後述するようにして各種情報が記憶される IC カードが挿入される。例えば楽曲のオーディオデータのダウンロードが行われたとすると、これに関する履歴情報が IC カードに記憶される。この IC カードの情報は、電話回線 4 を介して所定の機会、タイミングで課金サーバ 5 に送られる。課金サーバ 5 は、この送られてきた履歴情報に従って金額を設定して課金を行い、ユーザに請求する。

#### 【0024】

これまでの説明から分かるように、本発明が適用されたシステムでは、地上局 1 は、テレビ番組素材サーバ 6 からの音楽番組放送の素材となるビデオデータ及びオーディオデータと、楽曲素材サーバ 7 からのオーディオチャンネルの素材となるオーディオデータと、音声付加情報サーバ 8 からの音声データと、GUI データサーバ 9 からの GUI データとを多重化して送信している。

そして、各家庭の受信設備 3 でこの放送を受信すると、例えばモニタ装置 14 により、選局したチャンネルの番組を視聴することができる。また、番組のデータと共に送信される GUI データを利用した GUI 画面として、第 1 には EPG (Electrical Program Guide; 電子番組ガイド) 画面を表示させ、番組の検索等を行うことができる。また、第 2 には、例えば通常の番組放送以外の特定のサービス用の GUI 画面を利用して所要の操作を行うことで、本実施の形態の場合に

は、放送システムにおいて提供されている通常番組の視聴以外のサービスを楽しむことができる。

例えば、オーディオ（楽曲）データのダウンロードサービス用のGUI画面を表示させて、このGUI画面を利用して操作を行えば、ユーザが希望した楽曲のオーディオデータをダウンロードしてストレージデバイス13に記録して保存することが可能になる。

【0025】

なお、本実施の形態では、上記したようなGUI画面に対する操作を伴う、通常の番組放送以外の特定のサービスを提供するデータサービス放送については、インタラクティブ性を有することもあり、「インタラクティブ放送」ともいうことにする。

【0026】

## 1-2. GUI画面に対する操作

ここで、上述しているインタラクティブ放送の利用例、つまり、GUI画面に対する操作例について、図3及び図4を参照して概略的に説明しておく。ここでは、楽曲データ（オーディオデータ）のダウンロードを行う場合について述べる。

【0027】

先ず、図3によりIRD12に対してユーザが操作を行うためのリモートコントローラ64の操作キーについて、特に主要なものについて説明しておく。

図3には、リモートコントローラ64において各種キーが配列された操作パネル面が示されている。ここでは、これら各種キーのうち、電源キー101、数字キー102、画面表示切換キー103、インタラクティブ切換キー104、EPGキーパネル部105、チャンネルキー106について説明する。

【0028】

電源キー101は、IRD12の電源のオン／オフを行うためのキーである。数字キー102は、数字指定によりチャンネル切り換えを行ったり、例えばGU

I 画面において数値入力操作が必要な場合に操作するためのキーである。

画面表示切換キー 103 は、例えば通常の放送画面と EPG 画面との切り換えを行うキーである。例えば、画面表示切換キー 103 により EPG 画面を呼び出した状態の下で、EPG キーパネル部 105 に配置されたキーを操作すれば、電子番組ガイドの表示画面を利用した番組検索が行えることになる。また、EPG キーパネル部 105 内の矢印キー 105a は、後述するサービス用の GUI 画面におけるカーソル移動などにも使用することができる。

インタラクティブ切換キー 104 は、通常の放送画面と、その放送番組に付随したサービスのための GUI 画面との切り換えを行うために設けられる。

チャンネルキー 106 は、IRD 12 における選局チャンネルをそのチャンネル番号の昇順、降順に従って順次切り換えていくために設けられるキーである。

#### 【0029】

なお、本実施の形態のリモートコントローラ 64 としては、例えばモニタ装置 14 に対する各種操作も可能に構成されているものとされ、これに対応した各種キーも設けられているものであるが、ここでは、モニタ装置 14 に対応するキー等の説明は省略する。

#### 【0030】

次に、図 4 を参照して GUI 画面に対する操作の具体例について説明する。

受信設備 3 により放送を受信して所望のチャンネルを選局すると、モニタ装置 14 の表示画面には、図 4 (a) に示すように、テレビ番組素材サーバ 6 から提供された番組素材に基づく動画像が表示される。つまり、通常の番組内容が表示される。ここでは、例えば音楽番組が表示されているものとする。また、この音楽番組には楽曲のオーディオデータのダウンロードサービス（インタラクティブ放送）が付随されているものとする。

そして、この音楽番組が表示されている状態の下で、例えばユーザがリモートコントローラ 64 のインタラクティブ切換キー 104 を操作したとすると、表示画面は図 4 (b) に示すような、オーディオデータのダウンロードのための GUI 画面に切り替わる。



## 【0031】

このGUI画面においては、先ず、画面の左上部のテレビ番組表示エリア21Aに対して、図4(a)にて表示されていたテレビ番組素材サーバ6からのビデオデータによる画像が縮小化されて表示される。

また、画面の右上部には、オーディオチャンネルで放送されている各チャンネルの楽曲のリスト21Bが表示される。また、画面の左下にはテキスト表示エリア21Cとジャケット表示エリア21Dが表示される。さらに、画面の右側には歌詞表示ボタン22、プロフィール表示ボタン23、情報表示ボタン24、予約録音ボタン25、予約済一覧表示ボタン26、録音履歴表示ボタン27、およびダウンロードボタン28が表示される。

## 【0032】

ユーザは、このリスト21Bに表示されている楽曲名を見ながら、興味のある楽曲を探していく。そして、興味のある楽曲を見つけたらリモートコントローラ64の矢印キー105a（EPGキーパネル部105内）を操作して、その楽曲が表示されている位置にカーソルを合わせた後、エンター操作を行う（例えば矢印キー105aのセンター位置を押圧操作する）。

これによって、カーソルを合わせた楽曲を試聴することができる。すなわち、各オーディオチャンネルでは、所定の単位時間中、同一の楽曲が繰り返し放送されているので、テレビ番組表示エリア21Aの画面はそのまま、IRD12により上記操作により選択された楽曲のオーディオチャンネルに切り換えて音声出力することで、その楽曲を聞くことができる。この時、ジャケット表示エリア21Dにはその楽曲のMDジャケットの静止画像が表示される

## 【0033】

また、例えば上記の状態で歌詞表示ボタン22にカーソルを合わせ、エンター操作を行う（以下、ボタン表示にカーソルを合わせ、エンター操作を行うことを「ボタンを押す」という）と、テキスト表示エリア21Cに楽曲の歌詞がオーディオデータと同期したタイミングで表示される。同様に、プロフィール表示ボタン23あるいは情報表示ボタン24を押すと、楽曲に対応するアーティストのプロフィールあるいはコンサート情報などがテキスト表示エリア21Cに表示され

る。このように、ユーザは、現在どのような楽曲が配信されているのかを知ることができ、更に各楽曲についての詳細な情報を知ることができる。

#### 【0034】

ユーザは試聴した楽曲を購入したい場合には、ダウンロードボタン28を押す。ダウンロードボタン28が押されると、選択された楽曲のオーディオデータがダウンロードされ、ストレージデバイス13に記憶される。楽曲のオーディオデータと共に、その歌詞データ、アーティストのプロフィール情報、ジャケットの静止画データ等をダウンロードすることもできる。

そして、このようにして楽曲のオーディオデータがダウンロードされる毎に、その履歴情報がIRD12内のICカードに記憶される。ICカードに記憶された情報は、例えば1カ月に一度ずつ課金サーバ5により取り込みが行われ、ユーザに対してデータサービスの使用履歴に応じた課金が行われる。これによって、ダウンロードされる楽曲の著作権を保護することができることにもなる。

#### 【0035】

また、ユーザは予めダウンロードの予約を行いたい場合には、予約録音ボタン25を押す。このボタンを押すと、GUI画面の表示が切り換わり、予約が可能な楽曲のリストが画面全体に表示される。例えばこのリストは1時間単位、1週間単位、チャネル単位等で検索した楽曲を表示することが可能である。ユーザはこのリストの中からダウンロードの予約を行いたい楽曲を選択すると、その情報がIRD12内に登録される。そして、すでにダウンロードの予約を行った楽曲を確認したい場合には、予約済一覧表示ボタン26を押すことにより、画面全体に表示させることができる。このようにして予約された楽曲は、予約時刻になるとIRD12によりダウンロードされ、ストレージデバイス13に記憶される。

#### 【0036】

ユーザはダウンロードを行った楽曲について確認したい場合には、録音履歴ボタン27を押すことにより、既にダウンロードを行った楽曲のリストを画面全体に表示させることができる。

#### 【0037】

このように、本発明が適用されたシステムの受信設備3では、モニタ装置14

のGUI画面上に楽曲のリストが表示される。そして、このGUI画面上の表示にしたがって楽曲を選択するとその楽曲を試聴することができ、その楽曲の歌詞やアーティストのプロフィール等を知ることができる。さらに、楽曲のダウンロードとその予約、ダウンロードの履歴や予約済楽曲リストの表示等を行うことができる。

## 【0038】

詳しいことは後述するが、上記図4（b）に示すようなGUI画面の表示と、GUI画面に対するユーザの操作に応答したGUI画面上での表示変更、及び音声出力は、前述したMHEG方式に基づいたシナリオ記述により、オブジェクトの関係を規定することにより実現される。ここでいうオブジェクトとは、図4（b）に示された各ボタンに対応するパーツとしての画像データや各表示エリアに表示される素材データとなる。

そして、本明細書においては、このGUI画面のような、シナリオ記述によってオブジェクト間の関係が規定されることで、或る目的に従った情報の出力態様（画像表示や音声出力等）が実現される環境を「シーン」というものとする。また、1シーンを形成するオブジェクトとしては、シナリオ記述のファイル自体も含まれるものとする。

## 【0039】

以上、説明したように、本発明が適用されたデジタル衛星放送システムでは放送番組が配信されると共に、複数のオーディオチャンネルを使用して楽曲のオーディオデータが配信される。そして、配信されている楽曲のリスト等を使用して所望の楽曲を探し、そのオーディオデータをストレージデバイス13に簡単に保存することができる。

なお、デジタル衛星放送システムにおける番組提供以外のサービスとしては、上記した楽曲データのダウンロードの他にも各種考えられる。例えば、いわゆるテレビショッピングといわれる商品紹介番組を放送した上で、GUI画面としては購買契約が結べるようなものを用意することも考えられる。

## 【0040】

### 1-3. 地上局

これまで、本実施の形態としてのデジタル衛星放送システムの概要について説明したが、以降、このシステムについてより詳しい説明を行っていくこととする。そこで、先ず地上局1の構成について図5を参照して説明する。

#### 【0041】

なお、以降の説明にあたっては、次のことを前提とする。

本実施の形態では、地上局1から衛星2を介しての受信設備3への送信を行うのにあたり、DSM-CC(デジタル蓄積メディア・コマンド・アンド・コントロール; Digital Storage Media-Command and Control)プロトコルを採用する。

DSM-CC(MPEG-part 6)方式は、既に知られているように、例えば、何らかのネットワークを介して、デジタル蓄積メディア(DSM)に蓄積されたMPEG符号化ビットストリームを取り出し(Retrieve)たり、或いはDSMに対してストリームを蓄積(Store)するためのコマンドや制御方式を規定したものである。そして本実施の形態においては、このDSM-CC方式がデジタル衛星放送システムにおける伝送規格として採用されているものである。

そして、DSM-CC方式によりデータ放送サービス(例えばGUI画面など)のコンテンツ(オブジェクトの集合)を伝送するためには、コンテンツの記述形式を定義しておく必要がある。本実施の形態では、この記述形式の定義として先に述べたMPEGが採用されるものである。

#### 【0042】

図5に示す地上局1の構成において、テレビ番組素材登録システム31は、テレビ番組素材サーバ6から得られた素材データをAVサーバ35に登録する。この素材データはテレビ番組送出システム39に送られ、ここでビデオデータは例えばMPEG2方式で圧縮され、オーディオデータは、例えばMPEG2オーディオ方式によりパケット化される。テレビ番組送出システム39の出力はマルチプレクサ45に送られる。

#### 【0043】

また、楽曲素材登録システム32では、楽曲素材サーバ7からの素材データ、つまりオーディオデータを、MPEG2オーディオエンコーダ36A、及びAT

RACエンコーダ36Bに供給する。MPEG2オーディオエンコーダ36A、ATRACエンコーダ36Bでは、それぞれ供給されたオーディオデータについてエンコード処理（圧縮符号化）を行った後、MPEGオーディオサーバ40A及びATRACオーディオサーバ40Bに登録させる。

MPEGオーディオサーバ40Aに登録されたMPEGオーディオデータは、MPEGオーディオ送出システム43Aに伝送されてここでパケット化された後、マルチプレクサ45に伝送される。ATRACオーディオサーバ40Bに登録されたATRACデータは、ATRACオーディオ送出システム43Bに4倍速ATRACデータとして送られ、ここでパケット化されてマルチプレクサ45に送出される。

【0044】

また、音声付加情報登録システム33では、音声付加情報サーバ8からの素材データである音声付加情報を音声付加情報データベース37に登録する。この音声付加情報データベース37に登録された音声付加情報は、音声付加情報送出システム41に伝送され、同様にして、ここでパケット化されてマルチプレクサ45に伝送される。

【0045】

また、GUI用素材登録システム34では、GUIデータサーバ9からの素材データであるGUIデータを、GUI素材データベース38に登録する。

【0046】

GUI素材データベース38に登録されたGUI素材データは、GUIオーサリングシステム42に伝送され、ここで、GUI画面、即ち図4にて述べた「シーン」としての出力が可能なデータ形式となるように処理が施される。

【0047】

つまり、GUIオーサリングシステム42に伝送されてくるデータとしては、例えば、楽曲のダウンロードのためのGUI画面であれば、アルバムジャケットの静止画像データ、歌詞などのテキストデータ、更には、操作に応じて出力されるべき音声データなどである。

上記した各データはいわゆるモノメディアといわれるが、GUIオーサリング

システム42では、MHEGオーサリングツールを用いて、これらのモノメディアデータを符号化して、これをオブジェクトとして扱うようにする。

そして、例えば図4(b)にて説明したようなシーン(GUI画面)の表示態様と操作に応じた画像音声の出力態様が得られるように上記オブジェクトの関係を規定したシナリオ記述ファイル(スクリプト)と共にMHEG-5のコンテンツを作成する。

また、図4(b)に示したようなGUI画面では、テレビ番組素材サーバ6の素材データを基とする画像・音声データ(MPEGビデオデータ、MPEGオーディオデータ)と、楽曲素材サーバ7の楽曲素材データを基とするMPEGオーディオデータ等も、GUI画面に表示され、操作に応じた出力態様を与えられる。

従って、上記シナリオ記述ファイルとしては、上記GUIオーサリングシステム42では、上記したテレビ番組素材サーバ6の素材データを基とする画像・音声データ、楽曲素材サーバ7の楽曲素材データを基とするMPEGオーディオデータ、更には、音声付加情報サーバ8を基とする音声付加情報も必要に応じてオブジェクトとして扱われて、MHEGのスクリプトによる規定が行われる。

#### 【0048】

なお、GUIオーサリングシステム42から伝送されるMHEGコンテンツのデータとしては、スクリプトファイル、及びオブジェクトとしての各種静止画データファイルやテキストデータファイルなどとなるが、静止画データは、例えばJPEG(Joint Photograph Experts Group)方式で圧縮された640×480ピクセルのデータとされ、テキストデータは例えば800文字以内のファイルとされる。

#### 【0049】

GUIオーサリングシステム42にて得られたMHEGコンテンツのデータはDSM-CCエンコーダ44に伝送される。

DSM-CCエンコーダ44では、MPEG2フォーマットに従ったビデオ、オーディオデータのデータストリームに多重できる形式のトランスポートストリーム(以下TS(Transport Stream)とも略す)に変換して、パケット化されてマ

マルチプレクサ45に出力される。

【0050】

マルチプレクサ45においては、テレビ番組送出システム39からのビデオパケットおよびオーディオパケットと、MPEGオーディオ送出システム43Aからのオーディオパケットと、ATRACオーディオ送出システム43Bからの4倍速オーディオパケットと、音声付加情報送出システム41からの音声付加情報パケットと、GUIオーサリングシステム42からのGUIデータパケットとが時間軸多重化されると共に、キー情報サーバ10（図1）から出力されたキー情報に基づいて暗号化される。

【0051】

マルチプレクサ45の出力は電波送出システム46に伝送され、ここで例えば誤り訂正符号の付加、変調、及び周波数変換などの処理を施された後、アンテナから衛星2に向けて送信出力するようにされる。

【0052】

#### 1-4. 送信フォーマット

次に、DSM-CC方式に基づいて規定された本実施の形態の送信フォーマットについて説明する。

図6は、地上局1から衛星2に送信出力される際のデータの一例を示している。なお、前述したように、この図に示す各データは実際には時間軸多重化されているものである。また、この図では、図6に示すように、時刻t1から時刻t2の間が1つのイベントとされ、時刻t2から次のイベントとされる。ここでいうイベントとは、例えば音楽番組のチャンネルであれば、複数楽曲のラインナップの組を変更する単位であり、時間的には30分或いは1時間程度となる。

【0053】

図6に示すように、時刻t1から時刻t2のイベントでは、通常の動画の番組放送で、所定の内容A1を有する番組が放送されている。また、時刻t2から始めるイベントでは、内容A2としての番組が放送されている。この通常の番組で

放送されているのは動画と音声である。

【0054】

MPEGオーディオチャンネル(1)～(10)は、例えば、チャンネルCH1からCH10の10チャンネル分用意される。このとき、各オーディオチャンネルCH1, CH2, CH3・・・CH10では、1つのイベントが放送されている間は同一楽曲が繰り返し送信される。つまり、時刻 $t_1$ ～ $t_2$ のイベントの期間においては、オーディオチャンネルCH1では楽曲B1が繰り返し送信され、オーディオチャンネルCH2では楽曲C1が繰り返し送信され、以下同様に、オーディオチャンネルCH10では楽曲K1が繰り返し送信されることになる。これは、その下に示されている4倍速ATRACオーディオチャンネル(1)～(10)についても共通である。

【0055】

つまり、図6において、MPEGオーディオチャンネルと4倍速ATRACオーディオチャンネルのチャンネル番号である( )内の数字が同じものは同じ楽曲となる。また、音声付加情報のチャンネル番号である( )内の数字は、同じチャンネル番号を有するオーディオデータに付加されている音声付加情報である。更に、GUIデータとして伝送される静止画データやテキストデータも各チャンネルごとに形成されるものである。これらのデータは、図7(a)～(d)に示すようにMPEG2のトランスポートパケット内で時分割多重されて送信され、図7(e)～(h)に示すようにしてIRD12内では各データパケットのヘッダ情報を用いて再構築される。

【0056】

また、上記図6及び図7に示した送信データのうち、少なくとも、データサービス(インタラクティブ放送)に利用されるGUIデータは、DSM-CC方式に則って論理的には次のようにして形成されるものである。ここでは、DSM-CCエンコーダ44から出力されるトランスポートストリームのデータに限定して説明する。

【0057】

図8(a)に示すように、DSM-CC方式によって伝送される本実施の形態



のデータ放送サービスは、Service Gatewayという名称のルートディレクトリの中に全て含まれる。Service Gatewayに含まれるオブジェクトとしては、ディレクトリ(Directory)、ファイル(File)、ストリーム(Stream)、ストリームイベント(Stream Event)などの種類が存在する。

【0058】

これらのうち、ファイルは静止画像、音声、テキスト、更にはMHEGにより記述されたスクリプトなどの個々のデータファイルとされる。

ストリームは例えば、他のデータサービスやAVストリーム(TV番組素材としてのMPEGビデオデータ、オーディオデータ、楽曲素材としてのMPEGオーディオデータ、ATRACオーディオデータ等)にリンクする情報が含まれる。

また、ストリームイベントは、同じくリンクの情報と時刻情報が含まれる。

ディレクトリは相互に関連するデータをまとめるフォルダである。

【0059】

そして、DSM-CC方式では、図8(b)に示すようにして、これらの単位情報とService Gatewayをそれぞれオブジェクトという単位と捉え、それぞれをBIOPメッセージという形式に変換する。

なお、本発明に関わる説明では、ファイル、ストリーム、ストリームイベントの3つのオブジェクトの区別は本質的なものではないので、以下の説明ではこれらをファイルとしてのオブジェクトに代表させて説明する。

【0060】

そして、DSM-CC方式では、図8(c)に示すモジュールといわれるデータ単位を生成する。このモジュールは、図8(b)に示したBIOPメッセージ化されたオブジェクトを1つ以上含むようにされたうえで、BIOPヘッダが付加されて形成される可変長のデータ単位であり、後述する受信側における受信データのバッファリング単位となる。

また、DSM-CC方式としては、1モジュールを複数のオブジェクトにより形成する場合の、オブジェクト間の関係については特に規定、制限はされていない。

い。つまり、極端なことをいえば、全く関係の無いシーン間における2以上のオブジェクトにより1モジュールを形成したとしても、DSM-CC方式のもとの規定に何ら違反するものではない。

【0061】

このモジュールは、MPEG2フォーマットにより規定されるセクションといわれる形式で伝送するために、図8(d)に示すように、機械的に「ブロック」といわれる原則固定長のデータ単位に分割される。但し、モジュールにおける最後のブロックについては規定の固定長である必要はないものとされている。このように、ブロック分割を行うのはMPEG2フォーマットにおいて、1セクションが4KBを越えてはならないという規定があることに起因する。

また、この場合には上記ブロックとしてのデータ単位と、セクションとは同義なものとなる。

【0062】

このようにしてモジュールを分割して得たブロックは、図8(e)に示すようにしてヘッダが付加されてDDB(Download Data Block)というメッセージの形式に変換される。

【0063】

また、上記DDBへの変換と並行して、DSI(Download Server Initiate)及びDII(Download Indication Information)という制御メッセージが生成される。

上記DSI及びDIIは、受信側(IRD12)で受信データからモジュールを取得する際に必要となる情報であり、DSIは主として、次に説明するカルーセル(モジュール)の識別子、カルーセル全体に関連する情報(カルーセルが1回転する時間、カルーセル回転のタイムアウト値)等の情報を有する。また、データサービスのルートディレクトリ(Service Gateway)の所在を知るための情報も有する(オブジェクトカルーセル方式の場合)。

【0064】

DIIは、カルーセルに含まれるモジュールごとに対応する情報であり、モジュールごとのサイズ、バージョン、そのモジュールのタイムアウト値などの情報

を有する。

#### 【0065】

そして、図8（f）に示すように、上記DDB、DSI、DIIの3種類のメッセージをセクションのデータ単位に対応させて周期的に、かつ、繰り返し送出するようにされる。これにより、受信機側では例えば目的のGUI画面（シーン）を得るのに必要なオブジェクトが含まれているモジュールをいつでも受信できるようにされる。

本明細書では、このような伝送方式を回転木馬に例えて「カルーセル方式」といい、図8（f）に示すようにして模式的に表されるデータ伝送形態をカルーセルというものとする。

ここで、1カルーセルに含まれるモジュールとしては複数とされて構わない。例えば、1カルーセルにより1つのデータサービスに必要な複数のモジュールを伝送するようにしてもよいものである。

また、「カルーセル方式」としては、「データカルーセル方式」のレベルと「オブジェクトカルーセル方式」のレベルとに分けられる。特にオブジェクトカルーセル方式では、ファイル、ディレクトリ、ストリーム、サービスゲートウェイなどの属性を持つオブジェクトをデータとしてカルーセルを用いて転送する方式で、ディレクトリ構造を扱えることがデータカルーセル方式と大きく異なる。本実施の形態のシステムでは、オブジェクトカルーセル方式を採用するものとされる。

#### 【0066】

また、図9に、MHEG方式に則ったデータサービスとしてのファイル（MHEG application file）のディレクトリ構造例を示す。上述のようにオブジェクトカルーセル方式は、このディレクトリ構造を扱えることに特徴を有する。

通常、Service Domainの入り口となる（MHEG application file）は、必ず、Service Gatewayの直下にある、app0/startupというファイルとなる。

基本的には、Service Domain（Service Gateway

y)の下にapplication directory (app0, app1  
 ... appN)があり、その下にstartupといわれるアプリケーション  
 ・ファイルと、applicationを構成する各sceneのdirect  
 ory (scene0, scene1...)があるようにされる。更にsce  
 ne directoryの下には、MHEG scene fileとsce  
 neを構成する各content fileがおかれることとしている。

## 【0067】

また、上記のようにしてカルーセルにより送信されるGUIデータ、つまり、  
 図5のDSM-CCエンコーダ44から出力されるデータとしては、トランスポ  
 ートストリームの形態により出力される。このトランスポートストリームは例え  
 ば図10に示す構造を有する。

図10(a)には、トランスポートストリームが示されている。このトランス  
 ポートストリームとはMPEGシステムで定義されているビット列であり、図の  
 ように188バイトの固定長パケット（トランスポートパケット）の連結により  
 形成される。

## 【0068】

そして、各トランスポートパケットは、図10(b)に示すようにヘッダと特  
 定の個別パケットに付加情報を含めるためのアダプテーションフィールドとパケ  
 ットの内容（ビデオ／オーディオデータ等）を表すペイロード（データ領域）と  
 からなる。

## 【0069】

ヘッダは、例えば実際には4バイトとされ、図10(c)に示すように、先頭  
 には必ず同期バイトがあるようにされ、これより後ろの所定位置にそのパケット  
 の識別情報であるPID (Packet\_ID)、スクランブルの有無を示すス  
 クランブル制御情報、後続するアダプテーションフィールドやペイロードの有無  
 等を示すアダプテーションフィールド制御情報が格納されている。

## 【0070】

これらの制御情報に基づいて、受信装置側ではパケット単位でデスクランブル  
 を行い、また、デマルチプレクサによりビデオ／オーディオ／データ等の必要パ

ケットの分離・抽出を行うことができる。また、ビデオ／オーディオの同期再生の基準となる時刻情報を再生することもここで行うことができる。

#### 【0071】

また、これまでの説明から分かるように、1つのトランスポートストリームには複数チャンネル分の映像／音声／データの packets が多重されているが、それ以外に PSI (Program Specific Information) といわれる選局を司るための信号や、限定受信（個人の契約状況により有料チャンネルの受信可否を決定する受信機能）に必要な情報（EMM／ECM）、EPGなどのサービスを実現するための SI (Service Information) が同時に多重されている。ここでは、PSI について説明する。

#### 【0072】

PSI は、図11に示すようにして、4つのテーブルで構成されている。それぞれのテーブルは、セクション形式という MPEG System に準拠した形式で表されている。

図11 (a) には、NIT (Network Information Table) 及び CAT (Conditional Access Table) のテーブルが示されている。

NIT は、全キャリアに同一内容が多重されている。キャリアごとの伝送諸元（偏波面、キャリア周波数、畳み込みレート等）と、そこに多重されているチャンネルのリストが記述されている。NIT の PID としては、PID=0x0010とされている。

#### 【0073】

CAT もまた、全キャリアに同一内容が多重される。限定受信方式の識別と契約情報等の個別情報である EMM (Entitlement Management Message) packets の PID が記述されている。PID としては、PID=0x0001により示される。

#### 【0074】

図11 (b) には、キャリアごとに固有の内容を有する情報として、PAT が示される。PAT には、そのキャリア内のチャンネル情報と、各チャンネルの内容を表す PMT の PID が記述されている。PID としては、PID=0x0000により示される。

【0075】

また、キャリアにおけるチャンネルごとの情報として、図11(c)に示すPMT(Program Map Table)のテーブルを有する。

PMTは、チャンネル別の内容が多重されている。例えば、図11(d)に示すような、各チャンネルを構成するコンポーネント(ビデオ/オーディオ等)と、デスクランブルに必要なECM(Encryption Control Message)パケットのPIDが記述されているPMTのPIDは、PATにより指定される。

【0076】

1-5. IRD

続いて、受信設備3に備えられるIRD12の一構成例について図12を参照して説明する。

【0077】

この図に示すIRD12において、入力端子T1には、パラボラアンテナ11のLNB15により所定の周波数に変換された受信信号を入力してチューナ/フロントエンド部51に供給する。

チューナ/フロントエンド部51では、CPU(Central Processing Unit)80からの伝送諸元等を設定した設定信号に基づいて、この設定信号により決定されるキャリア(受信周波数)を受信して、例えばビタビ復調処理や誤り訂正処理等を施すことで、トランスポートストリームを得るようにされる。

チューナ/フロントエンド部51にて得られたトランスポートストリームは、デスクランブラ52に対して供給される。また、チューナ/フロントエンド部51では、トランスポートストリームからPSIのパケットを取得し、その選局情報を更新すると共に、トランスポートストリームにおける各チャンネルのコンポーネントPIDを得て、例えばCPU80に伝送する。CPU80では、取得したPIDを受信信号処理に利用することになる。

【0078】

デスクランブラ52では、ICカード65に記憶されているデスクランブルキ

ーデータをCPU80を介して受け取ると共に、CPU80によりPIDが設定される。そして、このデスクランブルキーデータとPIDとに基づいてデスクランブル処理を実行し、トランスポート部53に対して伝送する。

【0079】

トランスポート部53は、デマルチプレクサ70と、例えばDRAM等により構成されるキュー (Queue) 71とからなる。キュー (Queue) 71は、モジュール単位に対応した複数のメモリ領域が列となるようにして形成されているものとされ、例えば本実施の形態では、32列のメモリ領域が備えられる。つまり、最大で32モジュールの情報を同時に格納することができる。

【0080】

デマルチプレクサ70の概略的動作としては、CPU80のDeMUXドライバ82により設定されたフィルタ条件に従って、デスクランブラ52から供給されたトランスポートストリームから必要なトランスポートパケットを分離し、必要があればキュー71を作業領域として利用して、先に図7(e)～(h)により示したような形式のデータを得て、それぞれ必要な機能回路部に対して供給する。

デマルチプレクサ70にて分離されたMPEGビデオデータは、MPEG2ビデオデコーダ55に対して入力され、MPEGオーディオデータは、MPEGオーディオデコーダ54に対して入力される。これらデマルチプレクサ70により分離されたMPEGビデオ／オーディオデータの個別パケットは、PES (Packetized Elementary Stream) と呼ばれる形式でそれぞれのデコーダに入力される。

【0081】

また、トランスポートストリームにおけるMHEGコンテンツのデータについては、デマルチプレクサ70によりトランスポートストリームからトランスポートパケット単位で分離抽出されながらキュー71の所要のメモリ領域に書き込まれていくことで、モジュール単位にまとめられるようにして形成される。そして、このモジュール単位にまとめられたMHEGコンテンツのデータは、CPU80の制御によってデータバスを介して、メインメモリ90内のDSM-CCバッファ91に書き込まれて保持される。

## 【0082】

また、トランスポートストリームにおける4倍速ATRA Cデータ（圧縮オーディオデータ）も、例えばトランスポートパケット単位に必要なデータがデマルチプレクサ70により分離抽出されてIEEE1394インターフェイス60に対して出力される。また、IEEE1394インターフェイス60を介した場合には、オーディオデータの他、ビデオデータ及び各種コマンド信号等を送出することも可能とされる。

## 【0083】

PESとしての形式によるMPEGビデオデータが入力されたMPEG2ビデオデコーダ55では、メモリ55Aを作業領域として利用しながらMPEG2フォーマットに従って復号化処理を施す。復号化されたビデオデータは、表示処理部58に供給される。

## 【0084】

表示処理部58には、上記MPEG2ビデオデコーダ55から入力されたビデオデータと、後述するようにしてメインメモリ90のMH E Gバッファ92にて得られるデータサービス用のG U I画面等のビデオデータが入力される。表示処理部58では、このようにして入力されたビデオデータについて所要の信号処理を施して、所定のテレビジョン方式によるアナログオーディオ信号に変換してアナログビデオ出力端子T2に対して出力する。

これにより、アナログビデオ出力端子T2とモニタ装置14のビデオ入力端子とを接続することで、例えば先に図4に示したような表示が行われる。

## 【0085】

また、PESによるMPEGオーディオデータが入力されるMPEG2オーディオデコーダ54では、メモリ54Aを作業領域として利用しながらMPEG2フォーマットに従って復号化処理を施す。復号化されたオーディオデータは、D/Aコンバータ56及び光デジタル出力インターフェイス59に対して供給される。

## 【0086】

D/Aコンバータ56では、入力されたオーディオデータについてアナログ音



声信号に変換してスイッチ回路 57 に出力する。スイッチ回路 57 では、アナログオーディオ出力端子 T3 又は T4 の何れか一方に対してアナログ音声信号を出力するように信号経路の切換を行う。

ここでは、アナログオーディオ出力端子 T3 はモニタ装置 14 の音声入力端子と接続されるために設けられているものとされる。また、アナログオーディオ出力端子 T4 はダウンロードした楽曲をアナログ信号により出力するための端子とされる。

また、光デジタル出力インターフェイス 59 では、入力されたデジタルオーディオデータを光デジタル信号に変換して出力する。この場合、光デジタル出力インターフェイス 59 は、例えば IEC 958 に準拠する。

#### 【0087】

メインメモリ 90 は、CPU 80 が各種制御処理を行う際の作業領域として利用されるものである。そして、本実施の形態では、このメインメモリ 90 において、前述した DSM-CC バッファ 91 と、MHEG バッファ 92 としての領域が割り当てられるようになっている。

MHEG バッファ 92 には、MHEG 方式によるスクリプトの記述に従って生成された画像データ（例えば GUI 画面の画像データ）を生成するための作業領域とされ、ここで生成された画像データはバスラインを介して表示処理部 58 に供給される。

#### 【0088】

CPU 80 は、IRD 12 における全体制御を実行する。このなかには、デマルチプレクサ 70 におけるデータ分離抽出についての制御も含まれる。

また、獲得した MHEG コンテンツのデータについてデコード処理を施すことで、スクリプトの記述内容に従って GUI 画面（シーン）を構成して出力するための処理も実行する。

#### 【0089】

このため、本実施の形態の CPU 80 としては、主たる制御処理を実行する制御処理部 81 に加え、例えば少なくとも、DeMUX ドライバ 82、DSM-CC デコーダブロック 83、及び MHEG デコーダブロック 84 が備えられる。本

実施の形態では、このうち、少なくともDSM-CCデコーダブロック83及びMHEGデコーダブロック84については、ソフトウェアにより構成される。

DeMUXドライバ82は、入力されたトランスポートストリームのPIDに基づいてデマルチプレクサ70におけるフィルタ条件を設定する。

DSM-CCデコーダブロック83は、DSM-Managerとしての機能を有するものであり、DSM-CCバッファ91に格納されているモジュール単位のデータについて、MHEGコンテンツのデータに再構築する。また、MHEGデコーダブロック84からのアクセスに従って所要のDSM-CCデコード等に関連する処理を実行する。

#### 【0090】

MHEGデコーダブロック84は、DSM-CCデコーダブロック83により得られたMHEGコンテンツのデータ、つまり、DSM-CCバッファ91にて得られているMHEGコンテンツのデータにアクセスして、シーン出力のためのデコード処理を行う。つまり、そのMHEGコンテンツのスクリプトファイルにより規定されているオブジェクト間の関係を実現していくことで、シーンを形成するものである。この際、シーンとしてGUI画面を形成するのにあたっては、MHEGバッファ92を利用して、ここで、スクリプトファイルの内容に従ってGUI画面の画像データを生成するようにされる。

#### 【0091】

DSM-CCデコーダブロック83及びMHEGデコーダブロック84間のインターフェイスには、U-U API (DSM-CC U-U API (Application Portability Interface)) が採用される。

U-U APIは、例えばクライアント(MHEGデコーダブロック84)側がDSM Managerオブジェクト(DSMの機能を実現するサーバオブジェクト; DSM-CCデコーダブロック83)にアクセスするためのインターフェイスであり、カルーセルに含まれるService Gateway, Directory, File, Stream, Stream Eventなどの属性を有するオブジェクトをファイルシステムのようにして構造的にアクセスすることができるようにしたAPIとされる。

【0092】

このAPIを通じてカルーセルに含まれるオブジェクトへのアクセスを行うことで、カルーセルを使用するプログラム（クライアント）がカルーセル受信動作を関知することなく、バス名を使用してオブジェクトにアクセスすることが可能になる。

【0093】

また、このU-U APIは、下層のデータ転送方式に関わらず利用することが出来るように規定されたインターフェイスの集合であることから、このAPIを利用するプログラムは、U-U APIを提供するどのようなデータ転送方式においても利用できるという利点を有する。

【0094】

ここで、CPU80の制御によりトランスポートストリームから1シーンを形成するのに必要な目的のオブジェクトを抽出するための動作例について説明しておく。

【0095】

DSM-CCでは、トランスポートストリーム中のオブジェクトの所在を示すのにIOR(Interoperable Object Reference)が使用される。IORには、オブジェクトを見つけ出すためのカルーセルに対応する識別子、オブジェクトの含まれるモジュールの識別子（以下module\_\_idと表記）、1つのモジュール中でオブジェクトを特定する識別子（以下object\_\_keyと表記）のほか、オブジェクトの含まれるモジュールの情報を持つDIIを識別するためのタグ(association\_\_tag)情報を含んでいる。

また、モジュール情報を持つDIIには、1つ以上のモジュールそれぞれについてのmodule\_\_id、モジュールの大きさ、バージョンといった情報と、そのモジュールを識別するためのタグ(association\_\_tag)情報を含んでいる。

【0096】

トランスポートストリームから抜き出されたIORがCPU80において識別された場合に、そのIORで示されたオブジェクトを受信、分離して得るプロセ

スは、例えば次のようになる。

(Pr1) CPU80のDeMUXドライバ82では、IORのassociation\_tagと同じ値を持つエレメンタリーストリーム（以下ESと表記）を、カルーセルにおけるPMTのESループから探し出してPIDを得る。このPIDを持つESにDIIが含まれていることになる。

(Pr2) このPIDとtable\_id\_extensionとをフィルタ条件としてデマルチプレクサ70に対して設定する。これにより、デマルチプレクサ70では、DIIを分離してCPU80に対して出力する。

(Pr3) DIIの中で、先のIORに含まれていたmodule\_idに相当するモジュールのassociation\_tagを得る。

(Pr4) 上記association\_tagと同じ値を有するESを、PMTのESループ（カルーセル）から探し出し、PIDを得る。このPIDを有するESに目的とするモジュールが含まれる。

(Pr5) 上記PIDとmodule\_idとをフィルタ条件として設定して、デマルチプレクサ70によるフィルタリングを行う。このフィルタ条件に適合して分離抽出されたトランスポート packets がキュー71の所要のメモリ領域（列）に格納されていくことで、最終的には、目的のモジュールが形成される。

(Pr6) 先のIORに含まれていたobject\_keyに相当するオブジェクトをこのモジュールから抜き出す。これが目的とするオブジェクトになる。このモジュールから抜き出されたオブジェクトは、例えば、DSM-CCバッファ91の所定の領域に書き込みが行われる。

例えば、上記動作を繰り返し、目的とするオブジェクトを集めてDSM-CCバッファ91に格納していくことで、必要とされるシーンを形成するMHEGコンテンツが得られることになる。

#### 【0097】

マンマシンインターフェイス61では、リモートコントローラ64から送信されてきたコマンド信号を受信してCPU80に対して伝送する。CPU80では、受信したコマンド信号に応じた機器の動作が得られるように、所要の制御処理を実行する。

【0098】

ICカードスロット62にはICカード65が挿入される。そして、この挿入されたICカード65に対してCPU80によって情報の書き込み及び読み出しが行われる。

【0099】

モデム63は、電話回線4を介して課金サーバ5と接続されており、CPU80の制御によってIRD12と課金サーバ5との通信が行われるように制御される。

【0100】

ここで、上記構成によるIRD12におけるビデオ／オーディオソースの信号の流れを、図4により説明した表示形態に照らし合わせながら補足的に説明する。

図4(a)に示すようにして、通常の番組を出力する場合には、入力されたトランスポートストリームから必要な番組のMPEGビデオデータとMPEGオーディオデータとが抽出されて、それぞれ復号化処理が施される。そして、このビデオデータとMPEGオーディオデータが、それぞれアナログビデオ出力端子T2と、アナログオーディオ出力端子T3に出力されることで、モニタ装置14では、放送番組の画像表示と音声出力が行われる。

【0101】

また、図4(b)に示したGUI画面を出力する場合には、入力されたトランスポートストリームから、このGUI画面(シーン)に必要なMH EGコンテンツのデータをトランスポート部53により分離抽出してDSM-CCバッファ91に取り込む。そして、このデータを利用して、前述したようにDSM-CCデコーダブロック83及びMH EGデコーダブロック84が機能することで、MH EGバッファ92にてシーン(GUI画面)の画像データが作成される。そして、この画像データが表示処理部58を介してアナログビデオ出力端子T2に供給されることで、モニタ装置14にはGUI画面の表示が行われる。

【0102】

また、図4(b)に示したGUI画面上で楽曲のリスト21Bにより楽曲が選

択され、その楽曲のオーディオデータを試聴する場合には、この楽曲のMPEGオーディオデータがデマルチプレクサ70により得られる。そして、このMPEGオーディオデータが、MPEGオーディオデコーダ54、D/Aコンバータ、スイッチ回路57、アナログオーディオ出力端子T3を介してアナログ音声信号とされてモニタ装置14に対して出力される。

## 【0103】

また、図4(b)に示したGUI画面上でダウンロードボタン28が押されてオーディオデータをダウンロードする場合には、ダウンロードすべき楽曲のオーディオデータがデマルチプレクサ70により抽出されてアナログオーディオ出力端子T4、光デジタル出力インターフェイス59、またはIEEE1394インターフェイス60に出力される。

## 【0104】

ここで、特にIEEE1394インターフェイス60に対して、図2に示したIEEE1394対応のMDレコーダ/プレーヤ13Aが接続されている場合には、デマルチプレクサ70ではダウンロード楽曲の4倍速ATracデータが抽出され、IEEE1394インターフェイス60を介してMDレコーダ/プレーヤ13Aに装填されているディスクに対して記録が行われる。また、この際には、例えばJPEG方式で圧縮されたアルバムジャケットの静止画データ、歌詞やアーティストのプロフィールなどのテキストデータもデマルチプレクサ70においてトランスポートストリームから抽出され、IEEE1394インターフェイス60を介してMDレコーダ/プレーヤ13Aに転送される。MDレコーダ/プレーヤ13Aでは、装填されているディスクの所定の領域に対して、これら静止画データ、テキストデータを記録することができるようになっている。

## 【0105】

## 2. 本発明に至った背景

例えば、オブジェクトカールセル方式によるデジタルデータ放送の受信を行っている場合、その放送中においてカールセルに含まれる或るオブジェクトのバー

ジョンが更新されたとすると、その時点で、更新前のそのオブジェクトのデータは無効となり、新しいバージョンのオブジェクトにアクセスすることが可能な状態となる。

#### 【0106】

但し、DSM-CC方式においては、カルーセルにおけるオブジェクトのバージョンが更新されたタイミングでこれをクライアント側に通知するインターフェイスを備えていない。つまり、IRD12の場合であれば、DSM-CCデコーダブロック83側の受信データにおいて、或るオブジェクトのバージョンアップがあったとしても、これを直ちにMHEGデコーダブロック84側が知ることが出来ない。

#### 【0107】

ここで、クライアントが、現在シーン表示に使用しているオブジェクトのバージョンアップを通知されないまま、次にそのオブジェクトをサーバから読み出したような場合、以前のオブジェクトとは異なるデータが読み出されて、このオブジェクトにより表示が行われてしまうことになる。例えば、オブジェクトのバージョンアップに関わらず以前の表示状態を維持したいような場合には、このようなクライアント側の動作では不都合を招くことになる。

また、逆にオブジェクトのバージョンアップに対応して、現在のシーン表示の一部の内容を変更する必要があるような場合には、或る機会でもってその更新されたオブジェクトをサーバから読み出すまでは、シーンの表示が変更されないことになる。つまり、実際のオブジェクトのバージョンアップのタイミングに対して、シーンの表示内容の更新のタイミングが遅れることになる。

このようにして、クライアントがオブジェクトのバージョンアップを通知されないことで、何らかの不都合が生じることになる。

#### 【0108】

### 3. 本実施の形態のオブジェクト更新通知制御

#### 3-1. U-U APIの一般的処理

そこで、本実施の形態では、以降説明するようにして、クライアント（MHEGデコーダブロック84）側においてオブジェクトのバージョンアップを知ることが出来るようにして、これに対応した適切なMHEGデコード処理が得られるようにすることを目的とするものである。更には、バージョンアップされたオブジェクトを特定できるようにして、サーバ側からのオブジェクトの再度読み出しが効率的に行われるようにすることを目的とするものである。また、このようなオブジェクトのバージョンアップの通知のためのインターフェイスとしては、U-U APIに準拠させるように構成されるものである。

## 【0109】

ここで、本実施の形態としてのバージョンアップ通知のためのインターフェイスを説明するのに先だって、図18により、U-U APIのインターフェイスの一般的な一例を説明しておく。

ここでは、クライアントであるMHEGデコーダブロック84が、ストリーム再生を行う場合を例に挙げている。この図において○内に示す数は、MHEGデコーダブロック84及びDSM-CCデコーダブロック83の処理手順を示すものである。以下、この処理手順に従って説明を行う。また、以降の説明においては、MHEGデコーダブロック84をクライアント、DSM-CCデコーダブロック83についてはサーバということにする。

## 【0110】

（処理1） クライアントは、所要のタイミングでEvent::Subscribe("stream on")をサーバに対して伝送する。

Event::Subscribe("stream on")は、以降において、"stream on" イベントを受け取ることをサーバに宣言するインターフェイスである。

## 【0111】

（処理2） Event::Subscribe("stream on")が受信されると、サーバでは、stream onイベントに対応するイベント番号を返す。ここでは、Event#10を設定してクライアントに対して伝送している。



【0112】

(処理3) クライアントでは、イベント番号を獲得したら、Event::notifyをサーバへ出力する。

Event::notifyとは、クライアント側からサーバ側に対して、サーバ側において、何らかのイベントの発生があれば、その通知を要求するインターフェイスである。

【0113】

(処理4) 上記(処理3)のnotifyに対する応答処理として、図18のようにして或るタイミングで受信データにおいて"stream on"イベントが発生したら、サーバは、"stream on"イベントに対して設定したイベント番号であるEvent#10をクライアントに伝送する。

(処理5) クライアントでは、受信した上記Event#10により、"stream on"イベントが発生したことを知ることになる。そして、この場合には例えばストリーム再生のためのMHEGデコード処理を実行する。

【0114】

3-2. 第1例

次に、上記図18に示したU-U APIインターフェイスに基づく本実施の形態のオブジェクト更新通知制御について説明する。

前述のように、DSM-CCではカルーセル内のオブジェクトの更新があったことを、そのタイミングでクライアントに知らせるための情報を有してはいない。また、更新されたオブジェクトを直接的に識別可能な情報も伝送はされない。つまり、現状では、クライアント側は受信したオブジェクトに更新があったとされるタイミングでこれを知ることが出来ず、また、その更新があったオブジェクトを特定することも出来ないことになる。

【0115】

但し、DSM-CCでは、あるオブジェクトについて更新(バージョンアップ)が行われた場合、そのオブジェクトが含まれるモジュールもこれに伴ってバー

ジョンアップされるようになっている。そして、このモジュールのバージョンアップは、図8にて説明したD I Iの内容に反映される。即ち、D I Iにおけるモジュールのバージョン情報が変更される。本実施の形態ではこれを利用する。

#### 【0116】

そして、本実施の形態のオブジェクト更新通知制御として、その第1例においては、U-U APIのインターフェイスとして" D I I \_ C H A N G E D " イベントを追加する。この" D I I \_ C H A N G E D " イベントは、サーバ側において、その内容が更新された新規のD I Iメッセージを受信したことを意味する。そして、図13の(処理1)～(処理4)に示すようにしてオブジェクト更新通知制御を実行する。

#### 【0117】

(処理1) クライアントは、E v e n t : : S u b s c r i b e (" D I I \_ C H A N G E D ") をサーバに伝達する。

(処理2) E v e n t : : S u b s c r i b e (" D I I \_ C H A N G E D ") を受信したサーバは、" D I I \_ C H A N G E D " イベントに対して設定したイベント番号をクライアントに返す。ここでは、" D I I \_ C H A N G E D " イベントに対してE v e n t # 2を設定して返している。

#### 【0118】

(処理3) 上記イベント番号を獲得した後、クライアントは、E v e n t : : n o t i f yをサーバに伝送し、" D I I \_ C H A N G E D " を含む何らかのイベントが発生したらこれを通知する要求を行う。

(処理4) サーバでは、受信したカルーセルのデータに含まれるD I Iごとにそのバージョン値を保持しているようにされる。そして、上記E v e n t : : n o t i f yを受けて後の或るタイミングで、受信したカルーセルのデータに含まれるD I Iのバージョン値が変更された、つまりD I Iの内容に切り換えがあったとする。これは、そのD I Iが示すカルーセルのモジュールにおいて、特定は出来ないが何らかのオブジェクトについてバージョンアップがあったことを意味する。

このようにして、D I Iのバージョンアップ(内容切り換え)があった、つま

り” D I I \_ C H A N G E D ” イベントが発生すると、サーバでは、（処理3）の E v e n t : : n o t i f y に対する応答として、” D I I \_ C H A N G E D ” イベントに対して設定したイベント番号（E v e n t # 2）をクライアントに伝送するようにされる。

## 【0119】

これにより、クライアントでは、少なくとも現在放送中のデータサービスにおいてオブジェクトの変更があったことをほぼリアルタイムで知ることが出来る。

そして、この通知に従って、オブジェクトのバージョンアップに対応した、何らかの適切なMHEGデコード処理（シーンに関する出力制御等）を実行することが可能になる。

## 【0120】

また、U-U A P I の規定では、インターフェイスを実行する際に、何らかの情報を付加して伝達を行ってもよいこととされている。

そこで、（処理4）によりイベント番号（E v e n t # 2）をクライアントに送信する際、図のようにして、イベント番号と共にD I I の更新に対応するバージョンアップされたモジュールの識別情報（m o d u l e I d）を付加すれば、クライアント側ではオブジェクトの内容に変更のあったモジュールを特定することが可能となる。

この場合、クライアント側では、更新されたオブジェクトの特定まではされないため、現在関心のあるオブジェクト（例えば、クライアント側でG U I 画面表示などのために現在使用しているオブジェクトなどである）がバージョンアップされたモジュールに属しているかどうかまでは分からない。このため、例えば、クライアント側では、現在出力中のシーン（G U I 画面等）に必要なオブジェクトを全てD S M - C C バッファ91から再ロードするようにされる。このようにすれば、実際に更新されたオブジェクトは現在出力中のシーンの内容の変更には関連しない可能性はあるものの、更新されたオブジェクトは現在出力中のシーンの内容の変更に関わるのであれば、確実にシーン内容の変更を行うことができることになる。

## 【0121】

ところで、(処理3)として示したEvent::notifyは、前述のように、何らかのイベントが発生したことの通知をクライアントからサーバに要求するものであるが、実際には、或るイベントの発生によってサーバ側でEvent::notifyに応答した通知を行うと、この時点で、Event::notifyは無効となる。

このため、上記図13に示した処理の実際において、はじめに(処理2)としてEvent::Subscribe("DII\_CHANGED")を発行して以降、サーバ側での"DII\_CHANGED"イベントの発生を逃さずに通知させるためには、一旦発行したEvent::notifyに応答したイベント通知が行われたら、この後、直ちに次のEvent::notifyを発行するようにしている。つまり、サーバ側において、Event::notifyが無効とされている期間ができるだけ生じないようにするものである。

このようにすれば、データサービス放送中において、DIIの切り換えが行われるごとに、これを逃すことなくほぼリアルタイムで"DII\_CHANGED"イベントの発生があったことをクライアントに逐次通知することが可能になる。

## 【0122】

また、上記した(処理1)である、Event::Subscribe("DII\_CHANGED")をクライアントからサーバに伝達するインターフェイスは、データサービス放送中におけるDIIの切り換えを逐一逃さずに通知できるようにすることを考慮して、MHEGデコーダブロック84のプログラム(MHEGエンジン)が立ち上がって直ぐのタイミングと、カルーセルのデータ内容(放送内容)自体について切り換えが行われたときに実行するようにされる。

なお、MHEGデコーダブロック84のプログラムを立ち上げる場合とは、例えば、これまでデータサービス放送が付随されていない放送を受信していた状態から、例えばチャンネルの切り換えや番組の変更が行われることで、新たにデータサービス放送が付随している番組を受信したときなどとされる。

【0123】

## 3-3. 第2例

続いて、第2例としてのオブジェクト更新通知制御について図14及び図15を参照して説明する。この第2例では、クライアント側で変更のあったオブジェクトを特定することまでが可能となる。

【0124】

図14に示す(処理1)～(処理8)までによる第2例の処理手順としては次のようになる。

(処理1) 第2例においても、第1例の場合と同様に"DIICHANGED"イベントが使用され、先ずクライアントは、Event::Subscribe("DIICHANGED")をサーバに伝達する。

(処理2) Event::Subscribe("DIICHANGED")を受信したサーバは、"DIICHANGED"イベントに対して設定したイベント番号をクライアントに返す。ここでも、"DIICHANGED"イベントに対応するイベント番号としてはEvent#2を設定して返している。

。

【0125】

(処理3) 上記イベント番号を獲得した後、クライアントはVersion::get(objectId)のインターフェイスをサーバに伝送する。

Version::get(objectId)とは、クライアント側において関心のあるオブジェクトをサーバ側に伝えるためのインターフェイスである。このインターフェイスの引数(objectId)には、その関心のあるオブジェクトについてのobjectIdが示される。なお、ここでいう「関心のあるオブジェクト」とは、例えばクライアント(MHEGデコーダブロック84)にて現在シーン出力(GUI画面表示等)に使用しているオブジェクトなどをいうものである。

また、実際のVersion::get( )としては、次のようなAPIと

なる。

```
interface Version {
    void get (in ObjRef obj, out u_short moduleId);
};
```

(処理4) 上記Version::get(objectId)の受信に  
答して、サーバからはそのobjectIdにより示されるオブジェクトが属  
するモジュールの識別情報であるmoduleIdを返送する。

このmoduleIdの返送を受けて、クライアント側では送信したobje  
ctIdと、返送されたmoduleIdとを対応付けして記憶する。

#### 【0126】

上記(処理3)→(処理4)からなるインターフェイスは、クライアントにお  
いて現在関心があるとされるオブジェクトについて全て行われる。ここで、例え  
ば最後の(処理3)→(処理4)に該当する(処理5)→(処理6)が完了した  
とすると、クライアント側においては、図15に示すようなテーブル情報が得ら  
れている。つまり、[(処理3)→(処理4)]・・・[(処理5)→(処理6  
)]が実行されたことで、クライアント側で関心のある全てのオブジェクトごと  
に、そのobjectIdとmoduleIdとが対応付けされたテーブルが得  
られることになる。

そして、この後(処理7)に移行する。

#### 【0127】

(処理7) クライアントは、Event::notifyをサーバに伝送し  
、何らかのイベントが発生したらこれを通知する要求を行う。

(処理8) サーバにおいて上記Event::notifyに対する応答と  
して、DIIの内容に切り換えがあって"DII\_CHANGED"イベントが  
発生したとされると、サーバから、"DII\_CHANGED"イベントに対し  
て設定したイベント番号(Event # 2)をクライアントに伝送するのである

が、前述のように、U-U APIでは、或るイベントの送信に際して、何らかの情報を付加してもよいことが規定されている。そこで、この場合には、イベント番号 (Event # 2) に対して、バージョンアップしたDIIが対応するモジュール (つまりバージョンアップされたオブジェクトが属するとされるモジュール) のmoduleIdを付加して伝送するようにされる。

これは、次のようなAPIである。

```
module DSM {
    interface Event {
        struct StreamEvent {
            u_short eventId;    //id for DII_
CHANGED
            AppNPT rAppTime;    //not used
            sequence<u_short, 65535> moduleI
dList;    // (NEW)
        }
    }
}
```

#### 【0128】

このようにして、クライアント側でイベント番号 (Event # 2) とmoduleIdが得られると、クライアント側では、図15に示したテーブルについてテーブルサーチを行って、テーブルの中から、(処理8)により伝送されたmoduleIdと一致したmoduleIdを検索する。この検索されたmoduleIdに対応付けされたobjectIdにより示されるオブジェクトが、更新されたオブジェクトである可能性を有していることになるので、クライアントは、このオブジェクトを更新のあったオブジェクトとして特定するようにされる。

この場合、例えばテーブル中において、(処理8)により伝送されたmodu

leIdと一致したmoduleIdと対応づけられたobjectIdが複数存在する可能性もある。この場合、これら複数のobjectIdにより示されるオブジェクトのうちの全てが実際に更新されているとは限らないが、ここでは、これら複数のオブジェクトについて更新があったものと見なすようにされる。こうすれば、実際に更新のあったオブジェクトは確実に含まれるようにされる。

## 【0129】

また、第2例においても、DIIの切り換えを逃さずに"DII\_CHANGED" イベントのイベント番号+moduleIdの通知がクライアントに対して行われるようにするには、第1例の場合と同様に、(処理1)であるEvent::Subscribe("DII\_CHANGED")のクライアントからサーバへの伝達は、MHEGデコーダブロック84のプログラム立ち上げ直後、及びカルーセルの切り換えが有ったときに行われるようにすることが必要となる。

また、実際における(処理7)としてのEvent::notifyのサーバへの伝達も、これに応答した何らかのイベントの通知が行われたら、イベント番号+moduleIdの返送(処理8)が得られたら、直ちに次のEvent::notify(#2)の伝達を行うようにされる。

## 【0130】

このようにして、更新されたオブジェクトを知ることで、クライアントではオブジェクトのバージョンアップに対応した適切なMHEGデコード処理を実行することが出来るが、一例として次のような処理を実行することが可能である。

前述したように、MHEGデコーダブロック84は、MHEGの記述内容に従って、DSM-CCバッファ91から必要なオブジェクトのデータを読み出して、MHEGバッファ92にてGUI画面等のシーンのデータを形成して表示処理部58に出力する。

ここで、図14に示す処理によって、1又は複数の特定のオブジェクトについてバージョンアップしたことが特定されたとする。そして、これらのオブジェクトは、現在GUI画面形成のためのオブジェクトとして使用されており、この場合には、そのオブジェクトの内容の変更(バージョンアップ)があれば、出来る



だけこれを迅速にGUI画面に反映させる必要があるものとする。

このような場合、MHEGデコーダブロック84は、DSM-CCバッファ91に格納されているモジュール単位のデータの中から、バージョンアップが特定されたオブジェクトのデータのみを再ロードするようにされる。そして、MHEGバッファ92にて、そのオブジェクトの内容についてのみ入れ替えるようにして、新規のシーンのデータを生成して、表示処理部58に出力させるようにする。

#### 【0131】

例えば更新されたオブジェクトが特定できないが、シーンについて何らかの変更が有るという可能性のみが通知されるような場合（例えば第1例）では、例えばシーン内容の一部変更に対応する場合でさえ、そのシーンを形成するための全オブジェクトについて、DSM-CCバッファ91から再読み出しを行って、MHEGバッファ92上にてシーンの再構築を行う必要が生じることになる。

これに対して、第2例のようにして更新されたオブジェクトが特定される場合には、上記処理のようにして、MHEGデコーダブロック84は、シーン内容の一部変更に必要なとされるオブジェクトのみについて処理を行えばよいことになる。つまり、同じシーン内容の一部変更を行うのにも、MHEGデコーダブロック84の処理負担は軽いものとすることができる。

#### 【0132】

### 3-4. 第3例

続いて、第3例としてのオブジェクト更新通知制御について図16及び図17を参照して説明する。

この第3例によっても、クライアント側で変更のあったオブジェクトを特定することまでが可能となる。但し、第3例においては、先の第1例及び第2例において使用された“DII\_CHANGED”イベントは使用されない。

#### 【0133】

図16に示す第2例の手順としては次のようになる。

(処理1) クライアントは、現在関心のあるオブジェクトについて、UpdateEvent::subscribe(objectId)をサーバ側に伝達する。

UpdateEvent::subscribe(objectId)は、引数に示すobjectIdにより示されるオブジェクトが属するモジュールのバージョンアップが行われた場合のイベント(UpdateEvent)を受け取れることを宣言するものであり、次のようなAPIを呼ぶことになる。

```
interface UpdateEvent {
    void subscribe(in ObjRef obj, out
u_short up_eventId);
    void notify (in u_short up_eventId);
};
```

【0134】

(処理2) 上記UpdateEvent::subscribe(objectId)を受信したサーバでは、この受信したUpdateEvent::subscribe(objectId)に固有の識別情報である、UpdateEventIdを返送する。このUpdateEventIdは、サーバ側にてユニークに設定されるIdである。

【0135】

クライアントは、現在関心のある全オブジェクトについて、UpdateEvent::subscribe(objectId)の伝達(処理1)と、これに応答したUpdateEventIdの受信(処理2)を繰り返すようにされる。ここでは、(処理3)→(処理4)のプロセスが、最後のオブジェクトについての(処理1)→(処理2)に相当するものとしている。

【0136】

そして、関心のある全オブジェクトに対応して(処理1)→(処理2)・・・

(処理3) → (処理4) が完了したとされる段階では、クライアント (MHEGデコーダブロック84) では、図17(a)に示すテーブル情報が得られている。このテーブル情報は、(処理1) (又は(処理3)) としてサーバに伝送した `objectId`、つまり関心のあるオブジェクトについての `objectId` と、これに応答してサーバから得られた `UpdateEventId` との対応が示される。

また、サーバ (DSM-CCデコーダブロック83) 側では、図17(b)に示すテーブル情報が得られている。このテーブル情報は、(処理1) (又は(処理3)) により伝送された `objectId` に対応して設定した `UpdateEventId`、更には、上記 `objectId` により示されるオブジェクトが属するモジュールを特定する `moduleId`、及びこのモジュールのバージョンナンバ (DIIにより示される) とが対応付けされたものとなる。

#### 【0137】

(処理5) 上記のようにして (処理1) → (処理2) . . . (処理3) → (処理4) まだが完了されると、クライアントは、`UpdateEvent::notify` をサーバに伝送し、`UpdateEvent` (モジュールの更新) が発生したらこれを通知する要求を行う。

(処理6) サーバにおいて上記 `UpdateEvent::notify` を受けて以降は、受信したDIIが示すモジュールのバージョン値と、図17(b)に示すテーブルとを照合している。ここで、テーブルの或る `moduleId` に対応するバージョンナンバが、上記DIIが示すモジュールのバージョン値と異なっていれば、そのモジュールについてバージョンアップが有ったことが識別される。

このようにしてモジュールについてバージョンアップが有った場合、つまり `UpdateEvent` が発生したら、サーバは、テーブルを参照して、そのバージョンアップが有ったとされるモジュールの `moduleId` と対応付けされた `UpdateEventId` を付加して、次のようなAPIを呼び出して `UpdateEventId` を返送する。

```
interface UpdateEvent {
    struct VersionUpEvent {
        u_short eventId;    //id for DII_C
HANGED
    }
}
```

【0138】

上記（処理6）によりUpdateEventIdを受信して得たクライアント側では、この受信したUpdateEventIdと一致する、テーブル（図17（a））のUpdateEventIdにアクセスする。そして、テーブル上で、このアクセスされたUpdateEventIdに対応付けされたobjectIdを見ることにより、バージョンアップされたオブジェクトを特定することになる。

【0139】

このように、第3例によってもバージョンアップされたオブジェクトが特定されるので、先の第2例の場合と同様に、MHEGデコーダブロック84側の処理をより効率的なものとすることが可能になる。

そして、第3例の場合には次のような利点も有する。図17（a）に示したように、クライアント（MHEGデコーダブロック84）が有するテーブルは、サーバ側でユニークに設定されたUpdateEventIdに対してobjectIdを対応付けしたものである。従って、クライアント側ではEvent::notifyに応答したUpdateEventIdを獲得すれば、一義的にobjectIdを特定することが出来る。つまり、第3例ではクライアント側でバージョンアップされたオブジェクトの特定をするのにテーブルサーチを行う必要はないものであり、それだけクライアントの処理負担が軽減されるものである。

【0140】

また、第3例の場合にも、（処理5）としてのUpdateEvent::n

otifyのサーバへの伝達も、これに応答したUpdateEventIdの返送(処理6)が得られたら、直ちに次のUpdateEvent::otifyの伝達を行うようにして、バージョンアップしたモジュールの通知が逃さず行われるようにするものである。

なお、クライアント側でバージョンの更新について関心が無くなった場合(例えば、GUI画面表示が停止された)には、

```
interface UpdateEvent {
    void unsubscribe (up_eventId) ;
};
```

のAPIを発行する。これにより、サーバ側ではUpdateEvent通知のための管理情報を消去する。

#### 【0141】

また、本発明としては、データ伝送方式としてDSM-CC方式を採用し、MHGのクライアント-サーバ間のインターフェイスとしてU-U APIを採用した場合について説明しているが、これに限定されるものではなく、上記実施の形態において説明した送信フォーマットに準ずる伝送方式、及びインターフェイスであれば本発明の適用が可能とされる。また、本発明が適用されるシステムとしてもデジタル衛星放送システムに限定されるものではなく、例えばケーブルテレビジョンなどの放送や、インターネット等において適用することも可能である。

#### 【0142】

##### 【発明の効果】

以上説明したように本発明は、例えばDSM-CC方式のもとでオブジェクトカルーセルによりデータ伝送を行うシステムの受信側におけるデータ伝達制御方法として、カルーセル(循環データ単位)内におけるオブジェクトについて更新が行われたことを、カルーセルからデータを受信して保持するサーバ側から、このカルーセルのデータを使用して所要の機能を実現するクライアントに対して通知

できるようにしたことで、クライアント側では、この通知に従って、オブジェクトの更新に応答して実行すべき処理を迅速に或いは効率的に実行することが可能になるものである。

## 【0143】

そして本発明では、上記のようなオブジェクトの更新を通知するためのオブジェクト更新通知処理として、例えばU-U API等の既存のインターフェイスのもとで、オブジェクトの更新に応じてその内容が更新されるモジュールに関する制御情報(DII)の変更を意味する制御情報変更イベント(DII\_CHANGED)を設定し、この制御情報変更イベントを利用して、クライアント側からサーバ側に制御情報の変更があったことを通知するように構成される。つまり、敢えて特化されたインターフェイスの規格を採用することなく、既存のインターフェイスに準拠した上で、オブジェクトの更新を通知を実現することができ、それだけ汎用性が与えられることになる。

ここで、制御情報変更イベントに基づいて、サーバ側から制御情報変更イベントの発生を通知する際、その制御情報が対応するモジュールのIDを付加すれば、クライアント側では、単にカルーセル内でオブジェクトの更新があったことだけではなく、更新されたオブジェクトが属するモジュールまで特定することが出来るため、それだけ、データの再ロードなどを実行する際には、効率的な処理を実行することが可能になる。

## 【0144】

また、本発明の他のオブジェクト更新通知処理として、上記と同様に、U-U API等の既存のインターフェイスのもとで、DIIの変更を意味する制御情報変更イベント(DII\_CHANGED)を設定し、クライアント側で制御情報変更イベントの受け取りを宣言(Event::subscribe)した後、関心のあるオブジェクトのIDをサーバに伝えと共に、サーバからはこのオブジェクトが属するモジュールのIDをクライアントに送るようにする。そして、この後のクライアントの制御情報変更イベントの受け取り要求に応じて、サーバ側では、更新された制御情報のモジュールのIDを付加して、制御情報変更イベントがあったことを通知するように構成する。

これにより、クライアント側では、オブジェクトIDとモジュールIDとを対応付けしたテーブルを検索することで、ほぼ更新されたオブジェクトの特定まで行うことが可能になる。この場合、クライアント側では、例えばオブジェクト単位でデータを再ロードすればよく、更に処理負担が軽くなる。

## 【0145】

また、更に他のオブジェクト更新通知処理として、クライアント側では、関心のあるオブジェクトIDを付加してモジュールの更新イベントの受け取りをサーバに宣言 (`subscribe`) し、サーバでは、この `subscribe` に応答して、固有のモジュール更新イベントのIDを設定してクライアントに返送するというインターフェイスを実行するようにされる。

そして、この後のクライアントのモジュール更新イベントの受け取り要求に応じて、サーバ側では、更新された制御情報のモジュール更新イベントIDを付加してモジュール変更イベントを送信するように構成する。

この構成によっても、クライアント側では、オブジェクトIDとモジュール更新イベントIDとを対応付けしたテーブルを検索することで、ほぼ更新されたオブジェクトの特定まで行うことが可能になる。そして、この構成では、クライアント側では獲得したモジュール更新イベントIDと対応するオブジェクトIDが一義的に得られるため、更新されたオブジェクトを特定するのにテーブルサーチを行う必要が無く、更にクライアント側の処理負担は軽減される。

## 【0146】

また、サブスクライブイベント (`Event::subscribe`) の伝達は、クライアントのプログラムの立ち上げ時、又は、カルーセル自体の切り換えが行われたときに実行する、また、イベント通知要求 (`Event::notify`) は、サーバからのイベント通知要求に対する応答メッセージが得られた後において、直ちに実行されるようにすることで、クライアントのプログラムの立ち上げ以降においては、オブジェクトの更新の通知をほぼ逃さずに得ることが可能になり、例えば、放送側でのオブジェクトの更新にほぼ即応したGUI画面の内容変更が可能になるなど、信頼性が向上することになる。

【図面の簡単な説明】

【図 1】

本発明の実施の形態のデジタル衛星放送受信システムの構成例を示すブロック図である。

【図 2】

本実施の形態における受信設備の構築例を示すブロック図である。

【図 3】

IRDのためのリモートコントローラの外観を示す正面図である。

【図 4】

放送画面とGUI画面との切り換えを示す説明図である。

【図 5】

地上局の構成例を示すブロック図である。

【図 6】

地上局から送信されるデータを示すチャート図である。

【図 7】

送信データの時分割多重化構造を示す説明図である。

【図 8】

DSM-CCによる送信フォーマットを示す説明図である。

【図 9】

データサービスのディレクトリ構造の一例を示す説明図である。

【図 10】

トランスポートストリームのデータ構造図である。

【図 11】

PSIのテーブル構造を示す説明図である。

【図 12】

IRDの構成を示す説明図である。

【図 13】

第1例としてのオブジェクト更新通知制御を示す説明図である。



【図14】

第2例としてのオブジェクト更新通知制御を示す説明図である。

【図15】

第2例においてクライアント側で用意されるテーブル構造を示す説明図である。

【図16】

第3例としてのオブジェクト更新通知制御を示す説明図である。

【図17】

第3例においてクライアント側で用意されるテーブル構造を示す説明図である。

【図18】

U-U APIインターフェイスの一般的な制御動作例を示す説明図である。

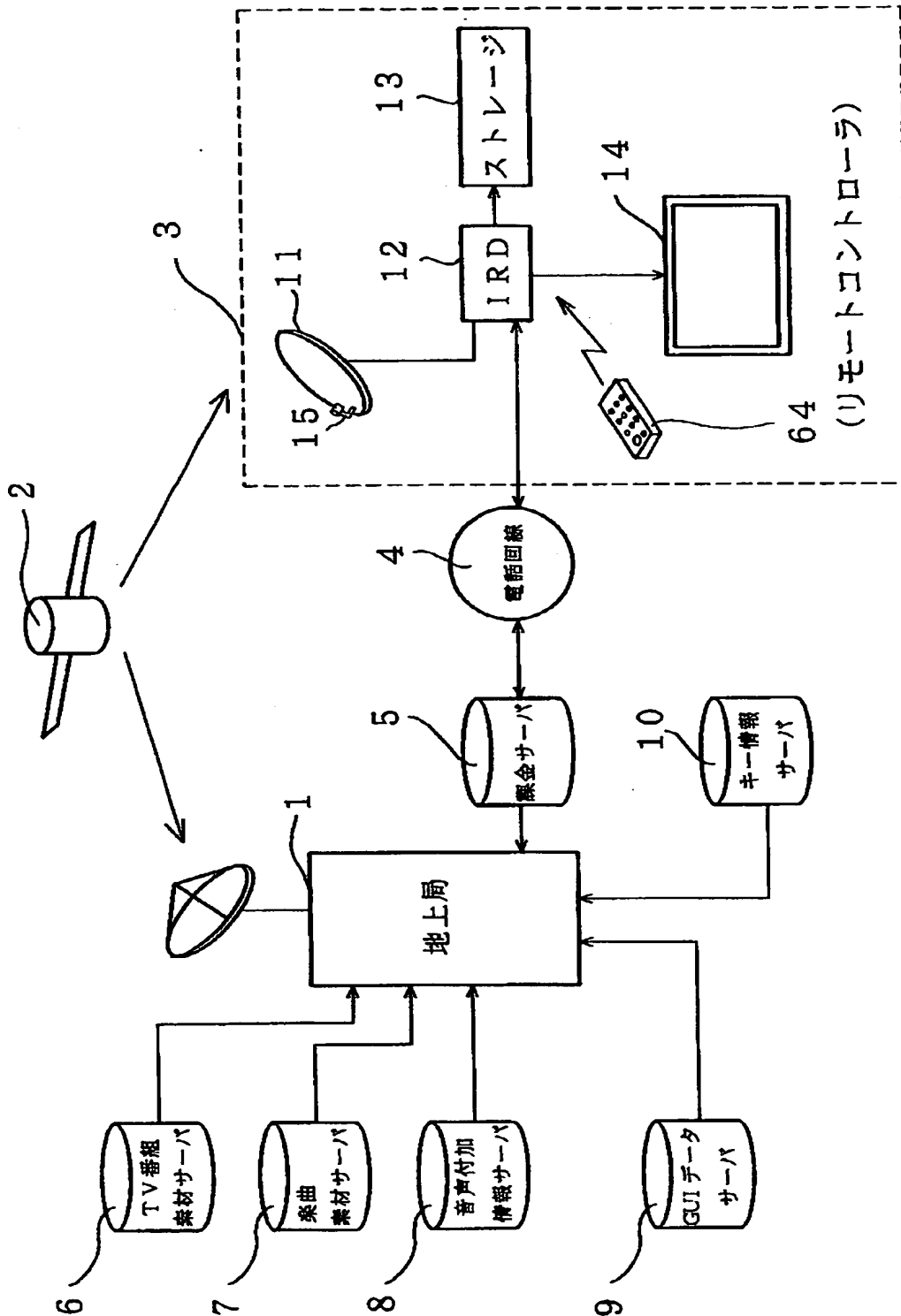
【符号の説明】

1 地上局、2 衛星、3 受信設備、5 課金サーバ、6 テレビ番組素材サーバ、7 楽曲素材サーバ、8 音声付加情報サーバ、9 GUIデータサーバ、10 キー情報サーバ、11 パラボラアンテナ、13 ストレージデバイス、13A MDレコーダ/プレーヤ、14 モニタ装置、16 IEEE1394バス、21A テレビ番組表示エリア、21B リスト、21C テキスト表示エリア、21D ジャケット表示エリア、22 歌詞表示ボタン、23 プロフィール表示ボタン、24 情報表示ボタン、25 予約録音ボタン、26 予約済一覧表示ボタン、27 録音履歴ボタン、28 ダウンロードボタン、31 テレビ番組素材登録システム、32 楽曲素材登録システム、33 音声付加情報登録システム、34 GUI用素材登録システム、35 AVサーバ、36A MPEGオーディオエンコーダ、36B ATRACエンコーダ、37 音声付加情報データベース、38 GUI素材データベース、39 テレビ番組送出システム、40A MPEGオーディオサーバ、40B MPEGオーディオサーバ、41 音声付加情報送出システム、42 GUIオーサリングシステム、43A MPEGオーディオ送出システム、43B ATRACオーディオ送出システム、44 DSM-CCエンコーダ、45 マルチプレクサ、46

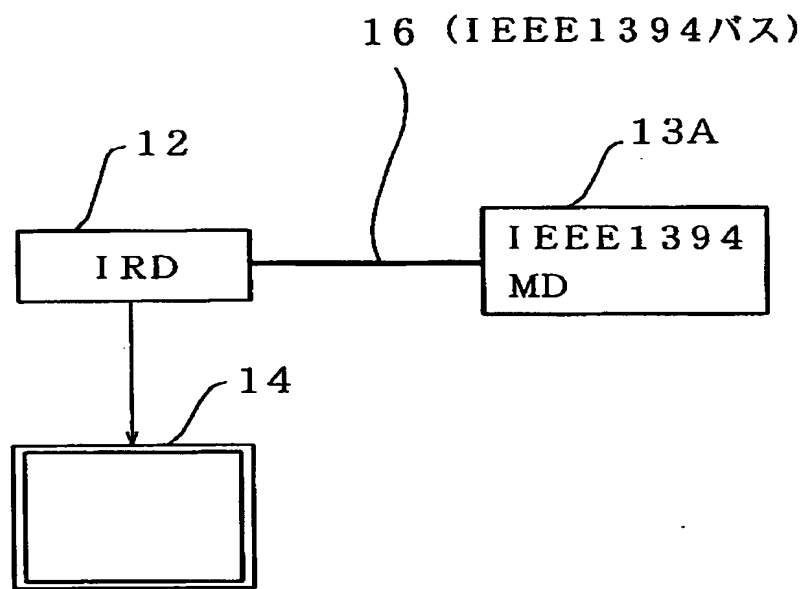
電波送出システム、51 チューナ／フロントエンド部、52 デスクランブラ、53 トランスポート部、54 MPEG2オーディオデコーダ、54A メモリ、55 MPEG2ビデオデコーダ、55A メモリ、56 D/Aコンバータ、57 スイッチ回路、58 表示処理部、59 光デジタル出力インターフェイス、60 IEEE1394インターフェイス、61 マンマシンインターフェイス、62 ICカードスロット、63 モデム、64 リモートコントローラ、65 ICカード、70 デマルチプレクサ、71 キュー、81 制御処理部、82 DeMUXドライバ、83 DSM-CCデコーダブロック、84 MHEGデコーダブロック、90 メインメモリ、91 DSM-CCバッファ、101 電源キー、102 数字キー、103 画面表示切換キー、104 インタラクティブ切換キー、105a 矢印キー、105 EPGキーパネル部、106 チャンネルキー、T1 入力端子、T2 アナログビデオ出力端子、T3 アナログオーディオ出力端子、T4 アナログオーディオ出力端子

【書類名】 図面

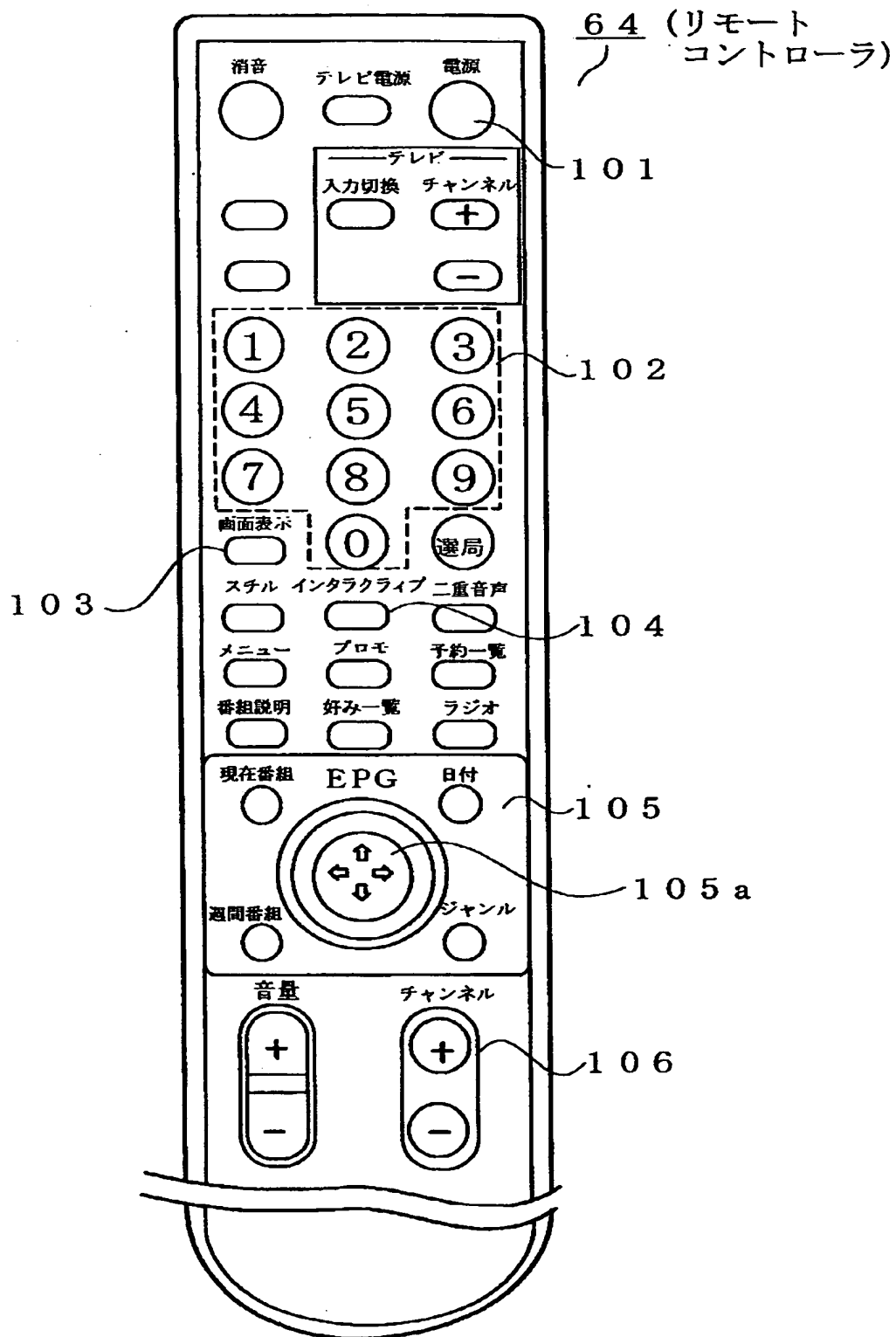
【図 1】



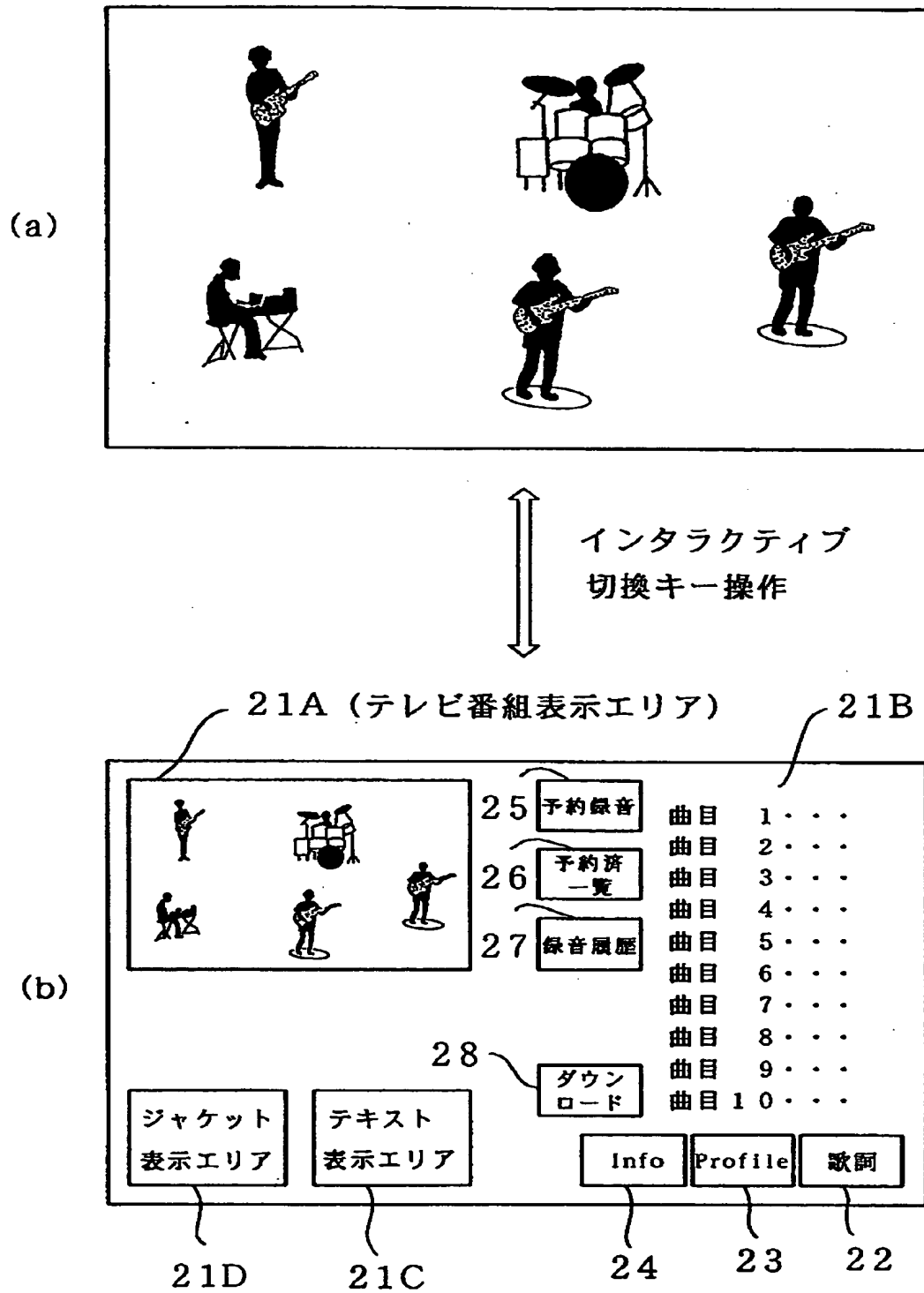
【図 2】



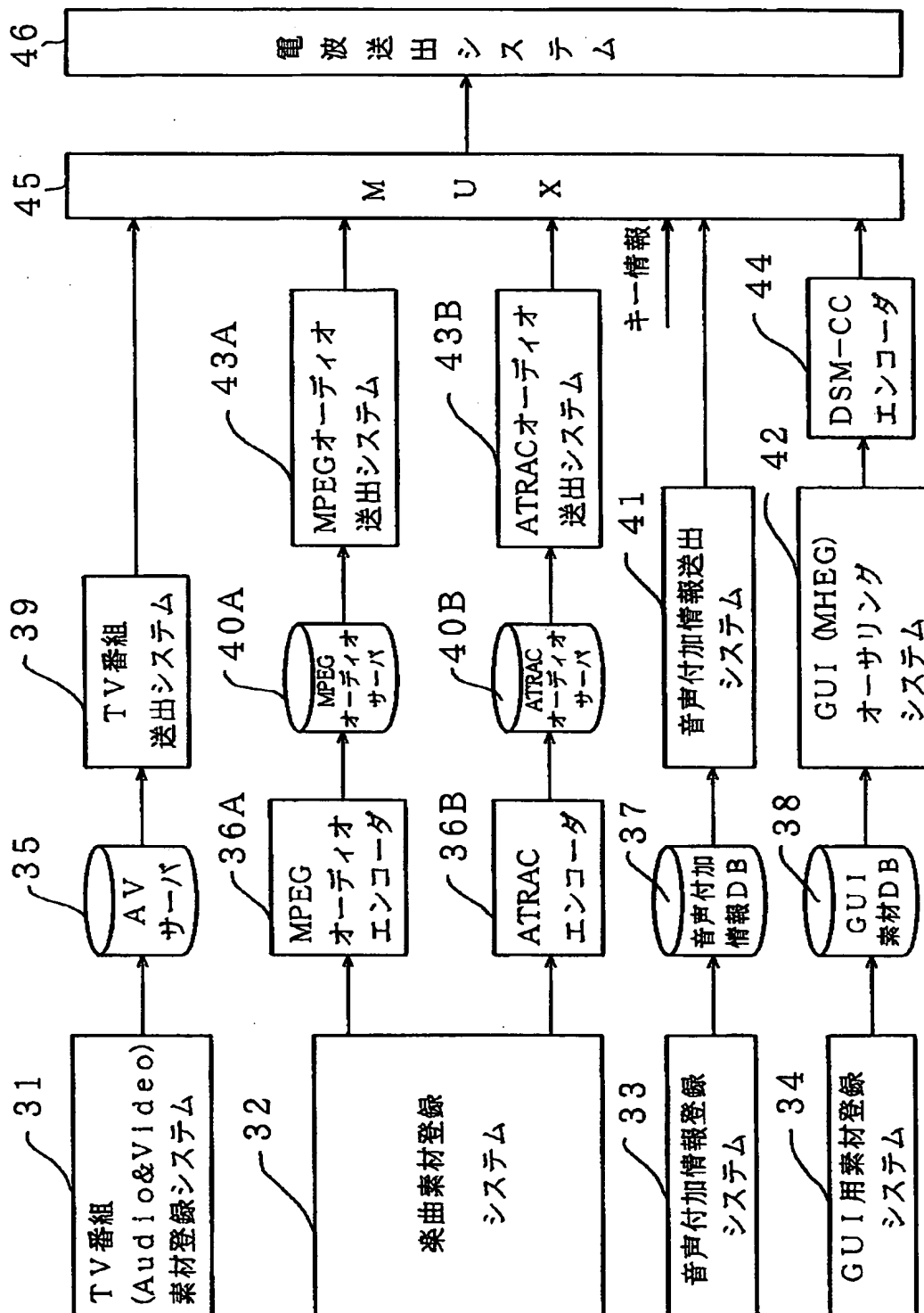
【図 3】



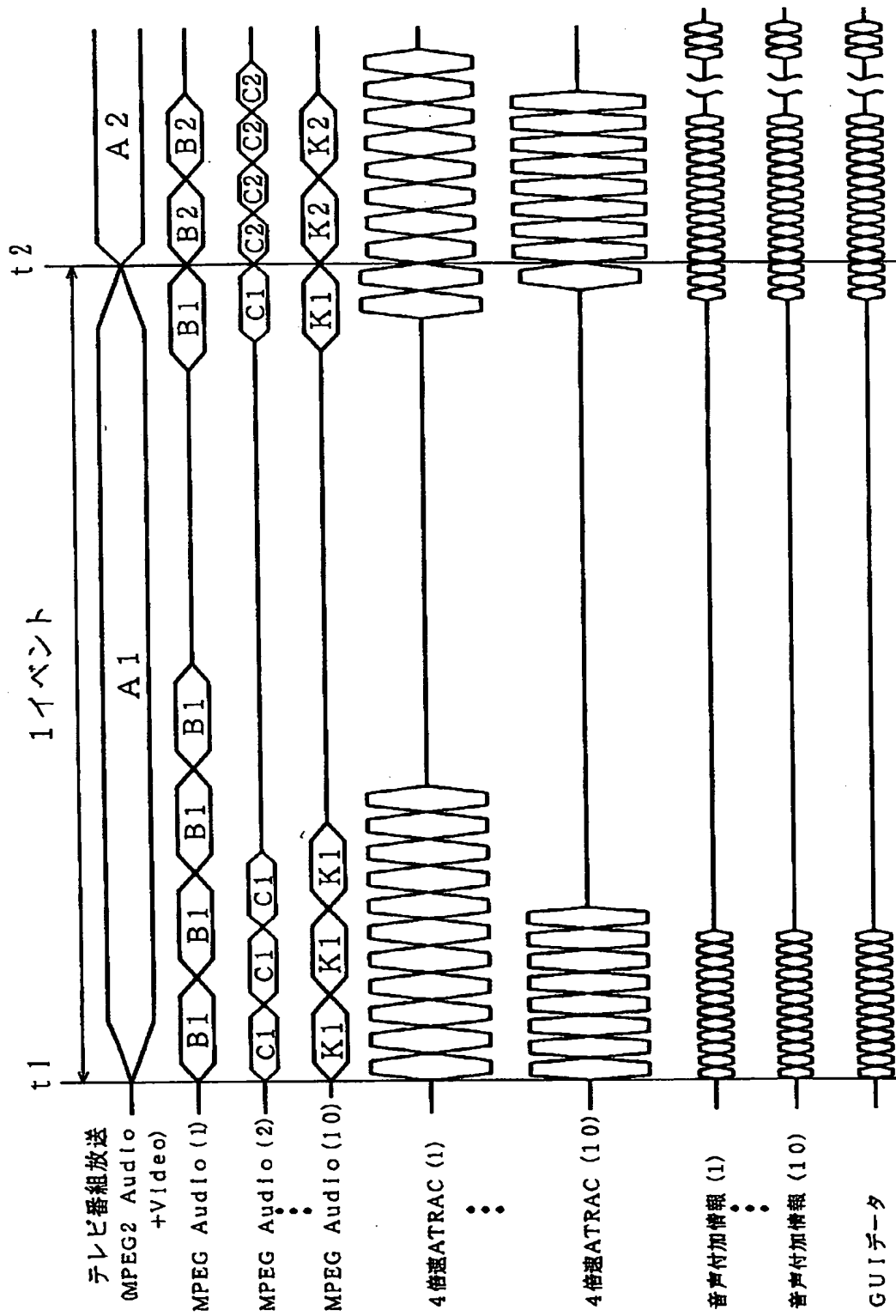
【図 4】



【図 5】

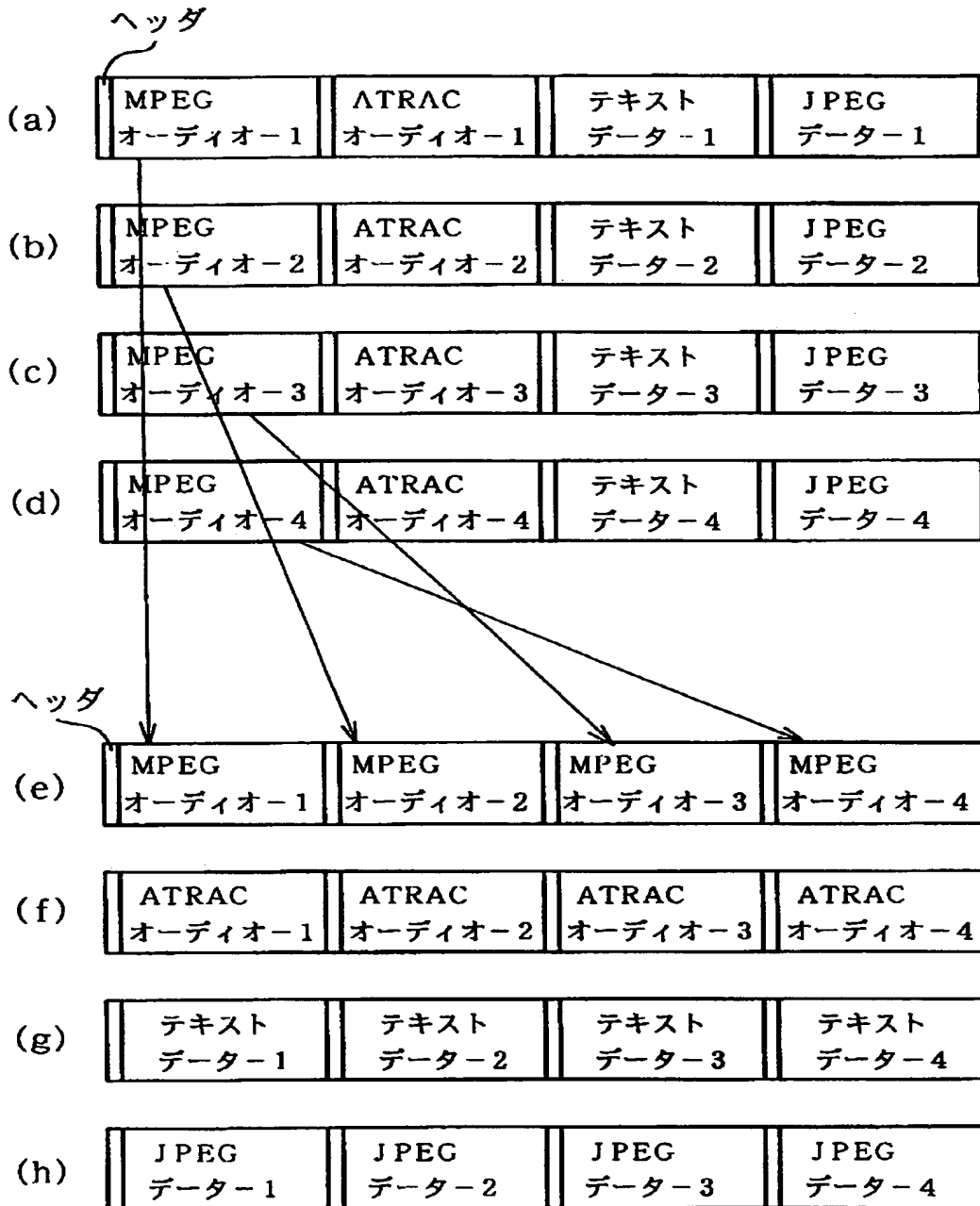


【図 6】



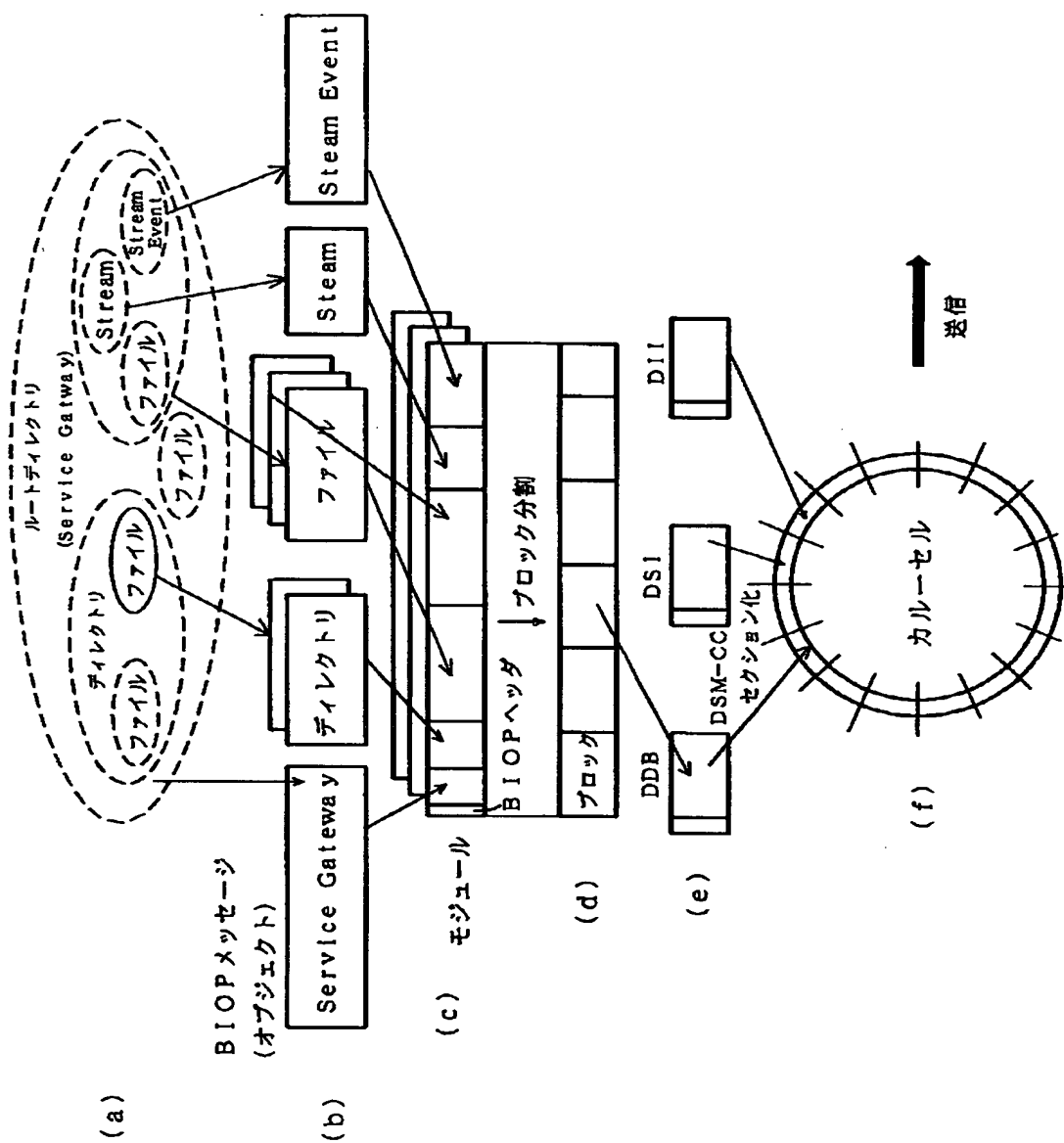


【図 7】

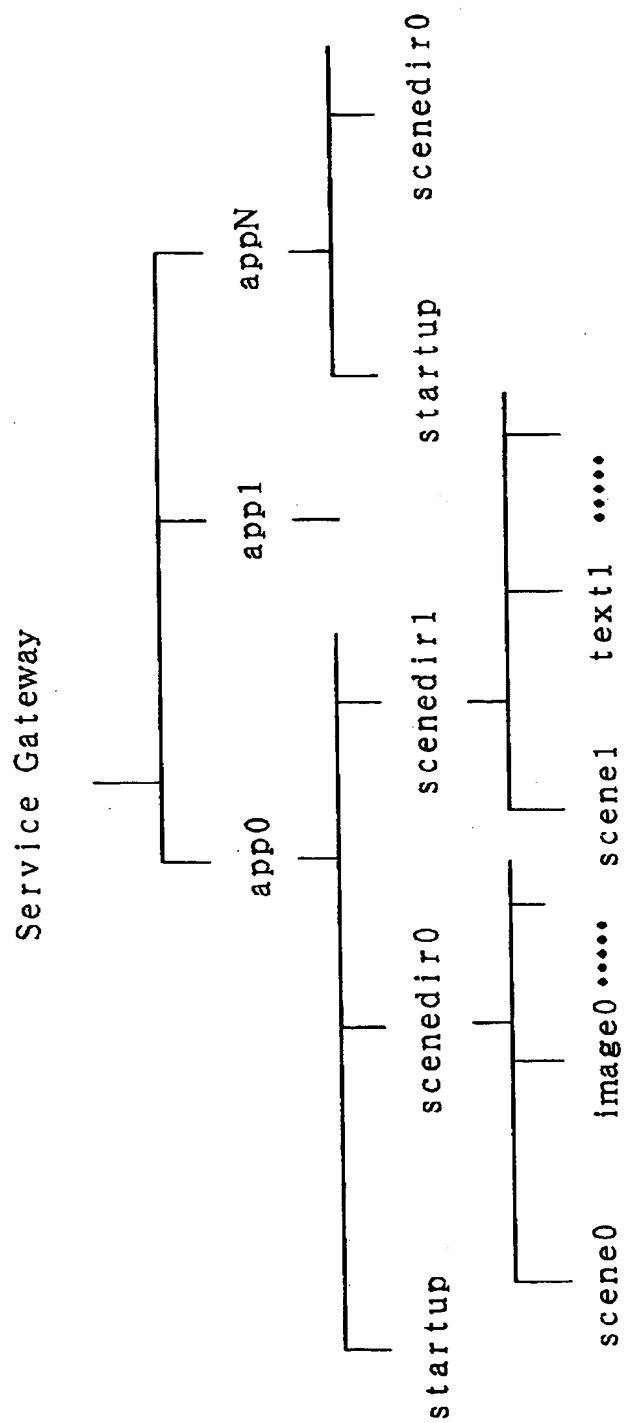


送信側の時分割多重化

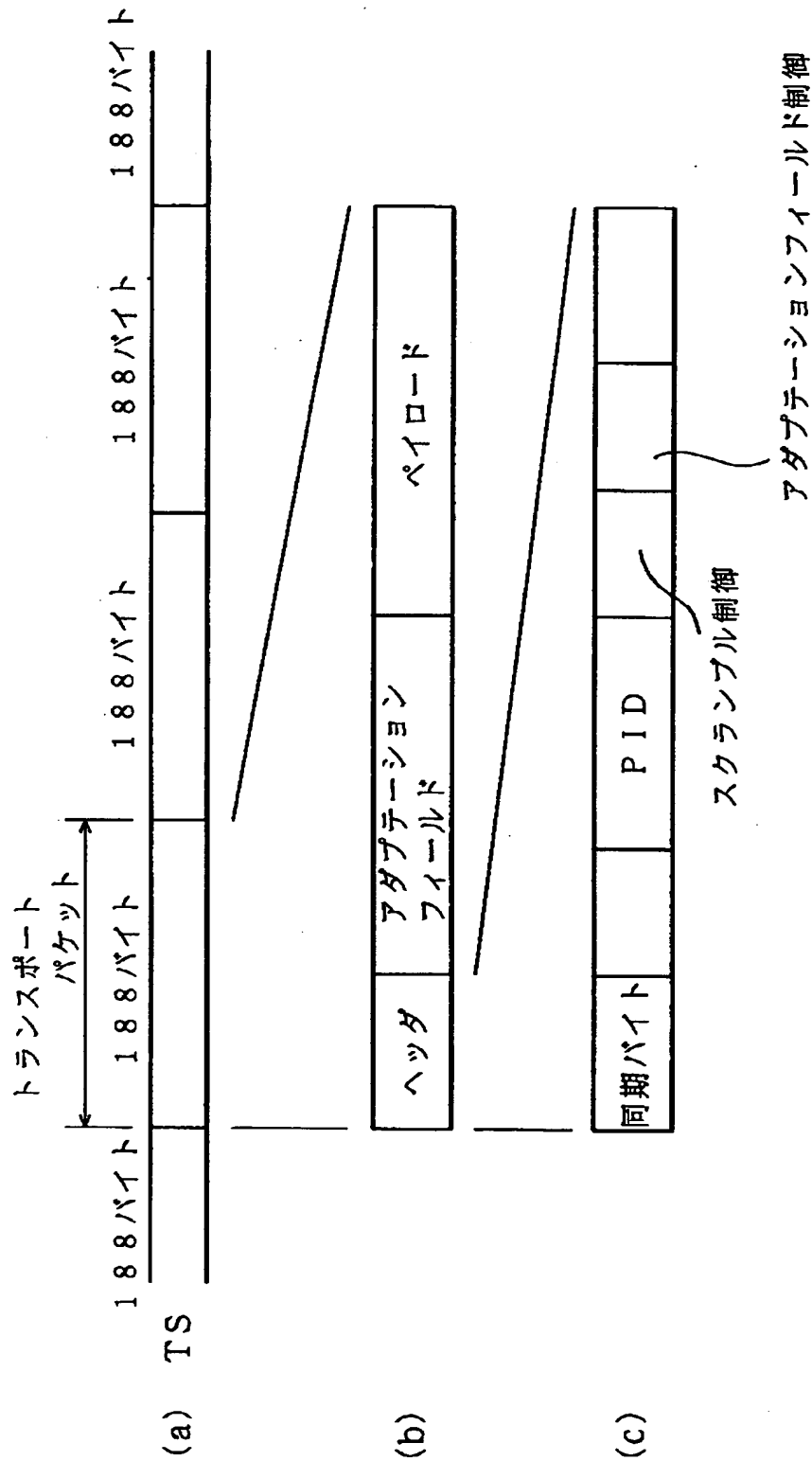
【図 8】



【図9】

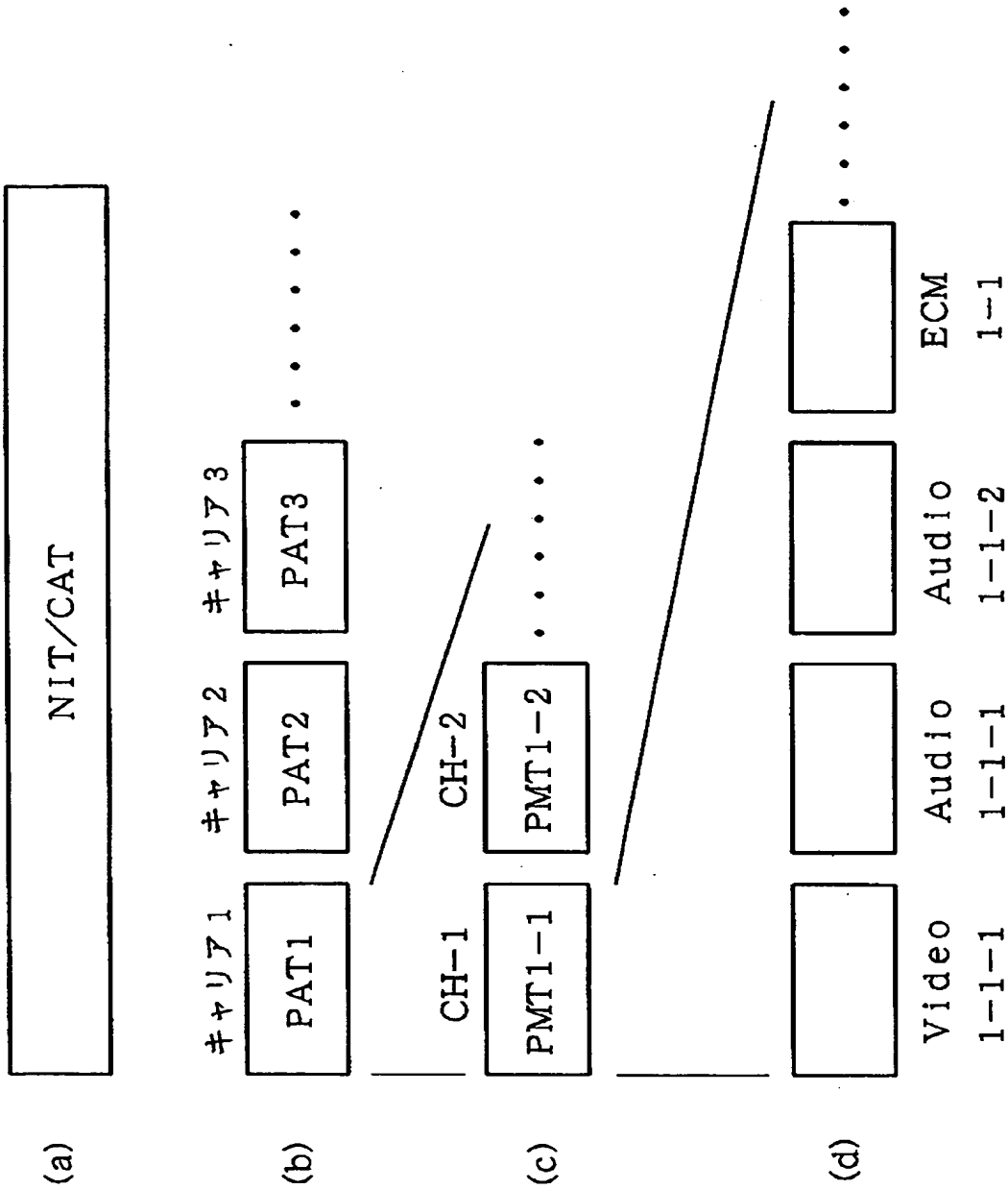


【図 10】

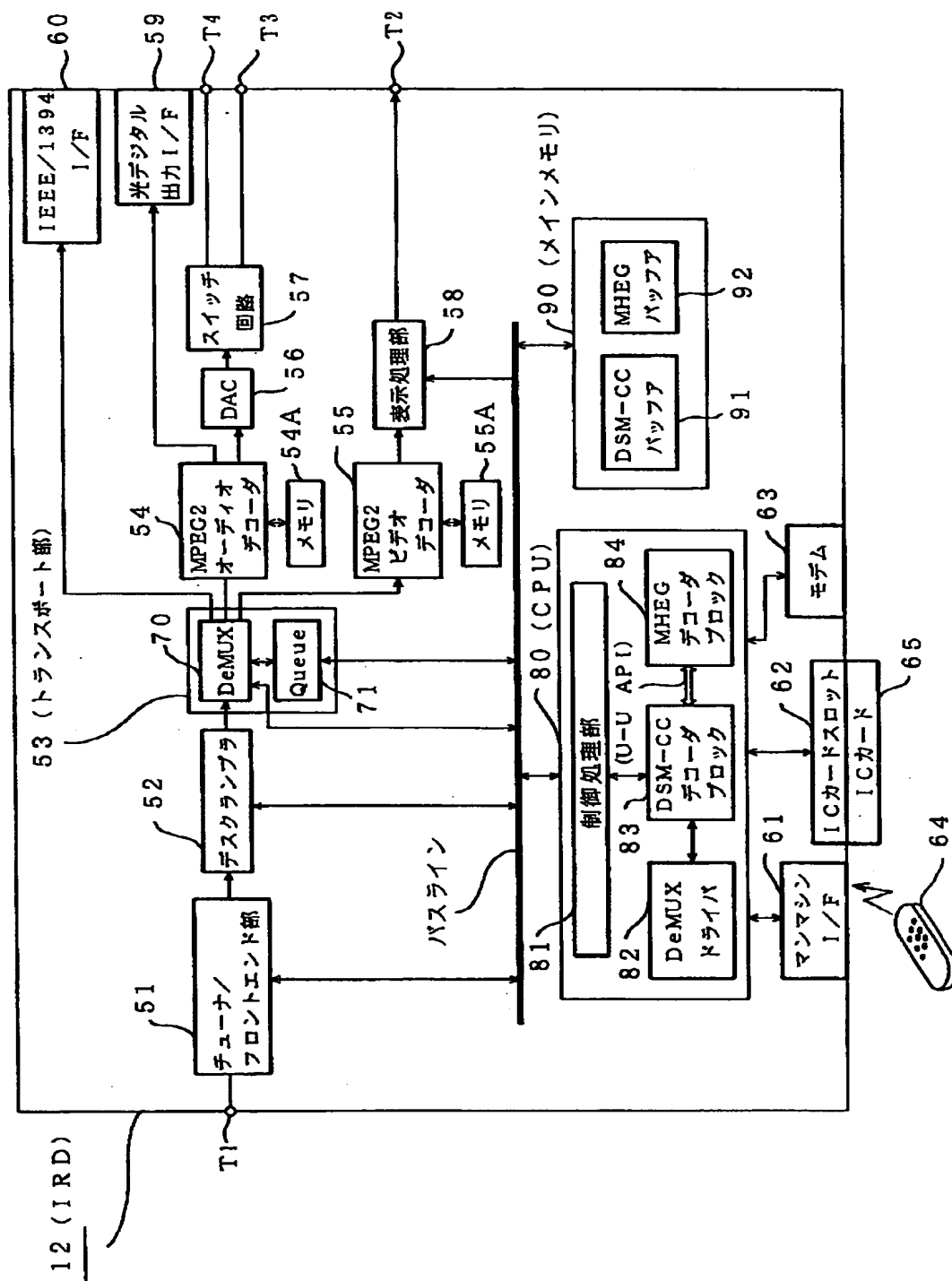


トランスポートパケット

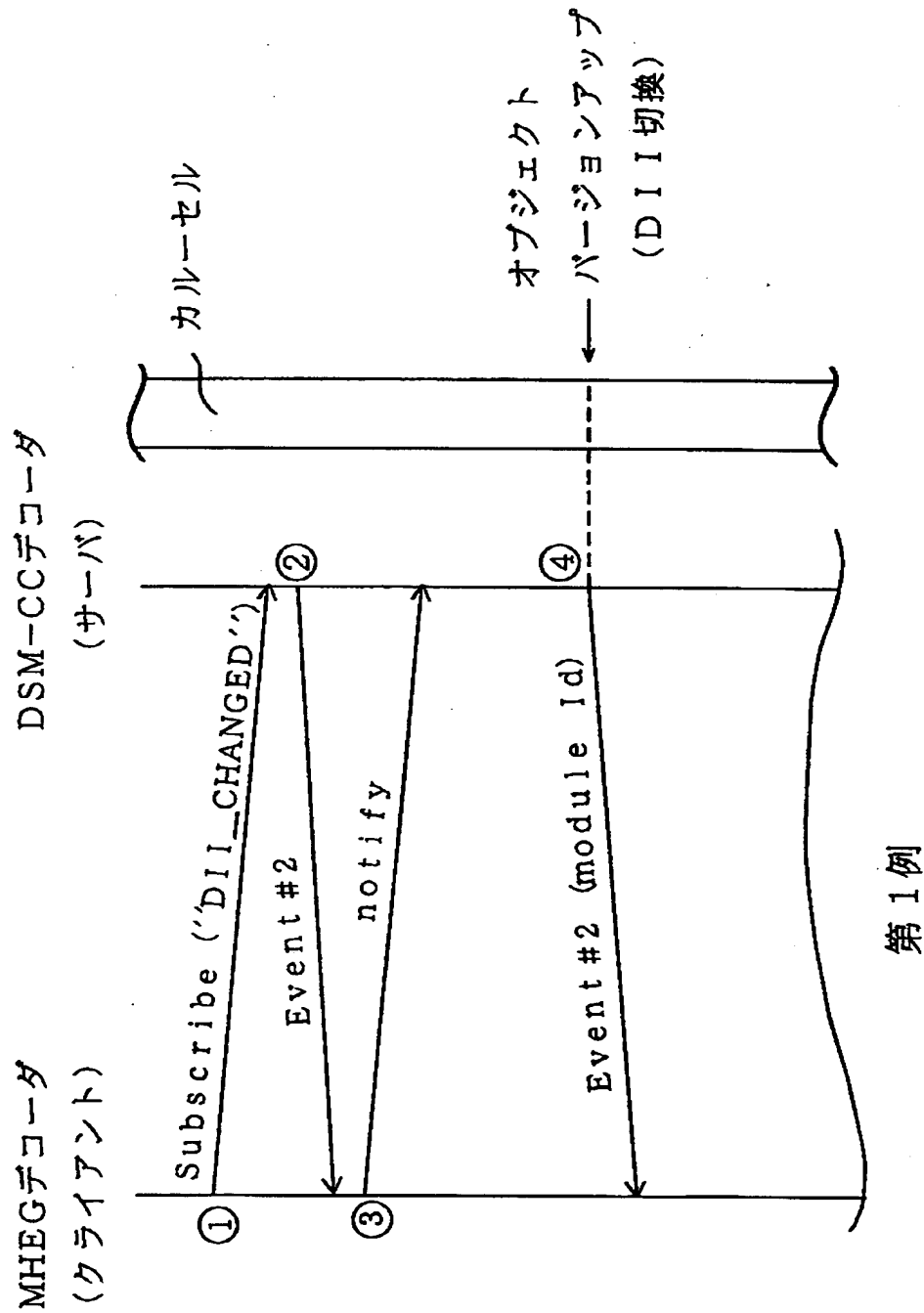
【図 11】



【図 12】

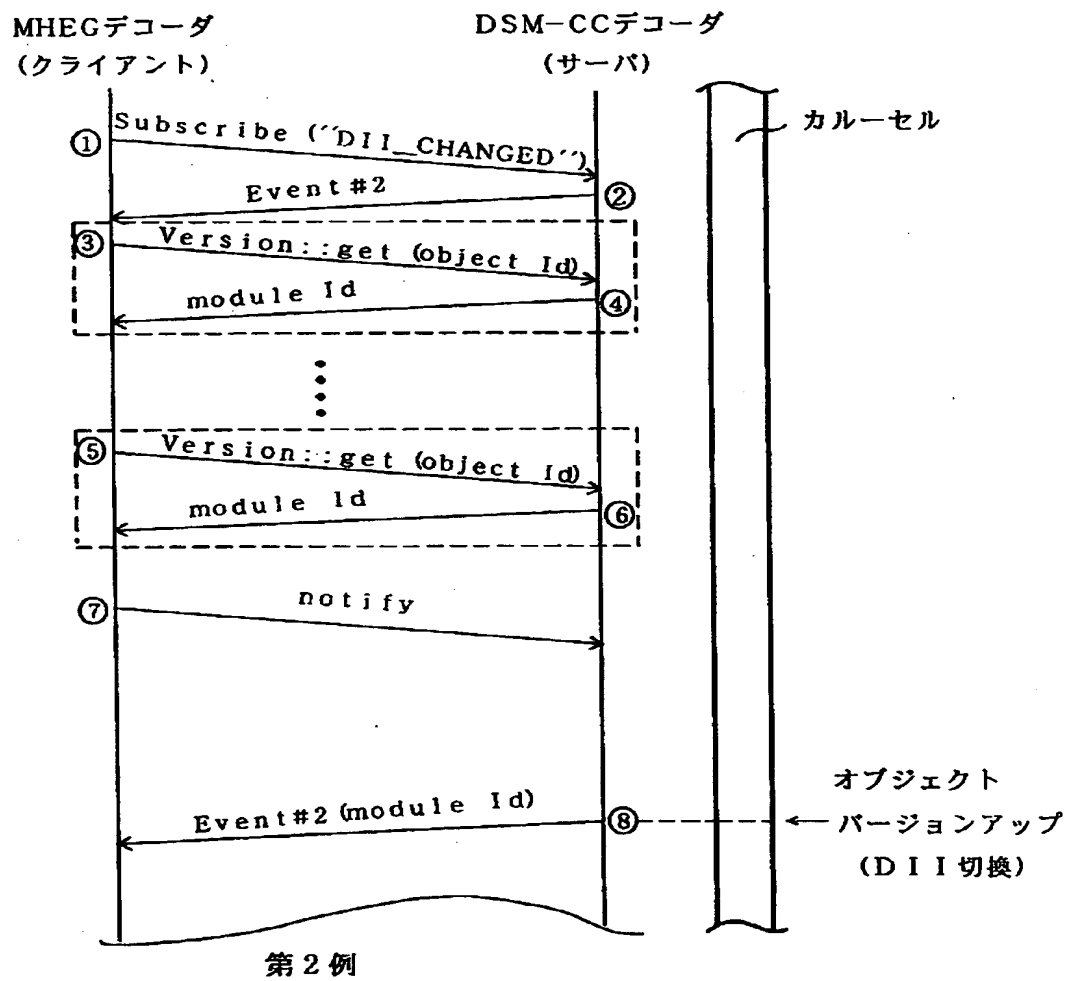


【図 13】



第 1 例

【図14】



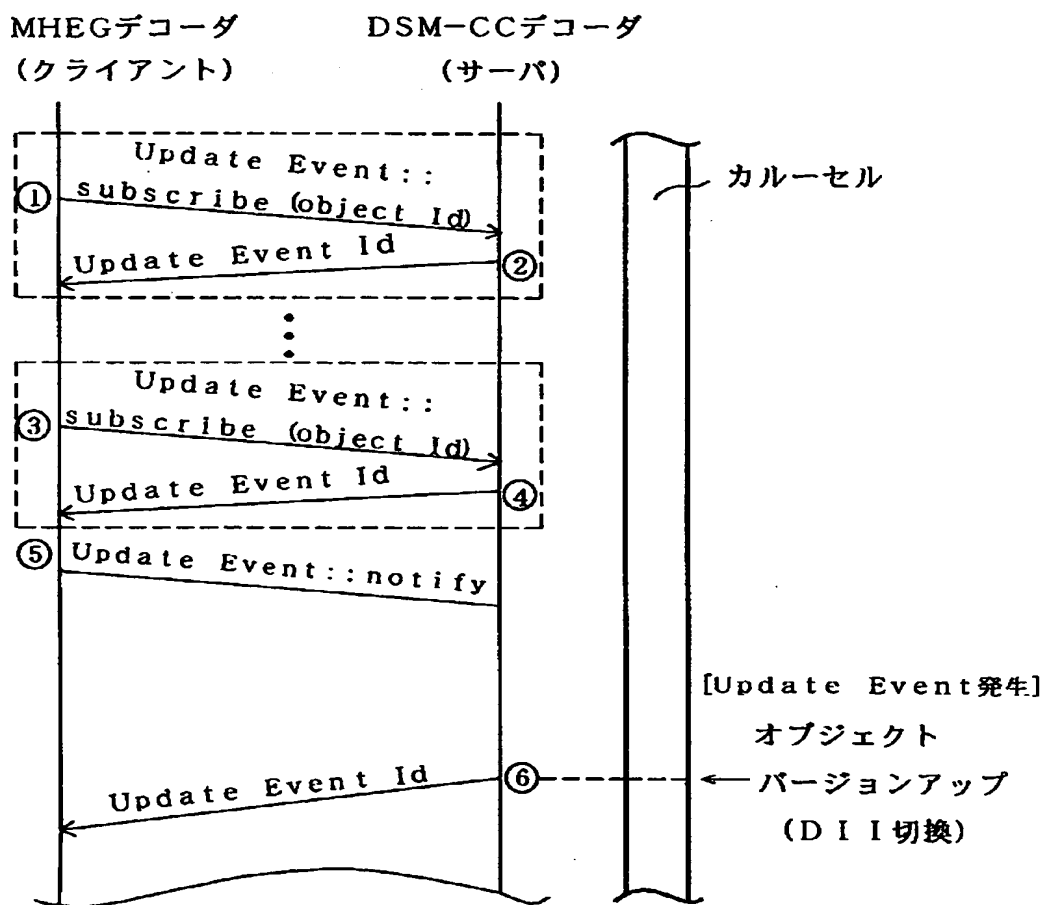


【図 15】

object Id	module Id

MHEGデコーダテーブル

【図16】



【図 17】

(a)

object Id	Update Event Id

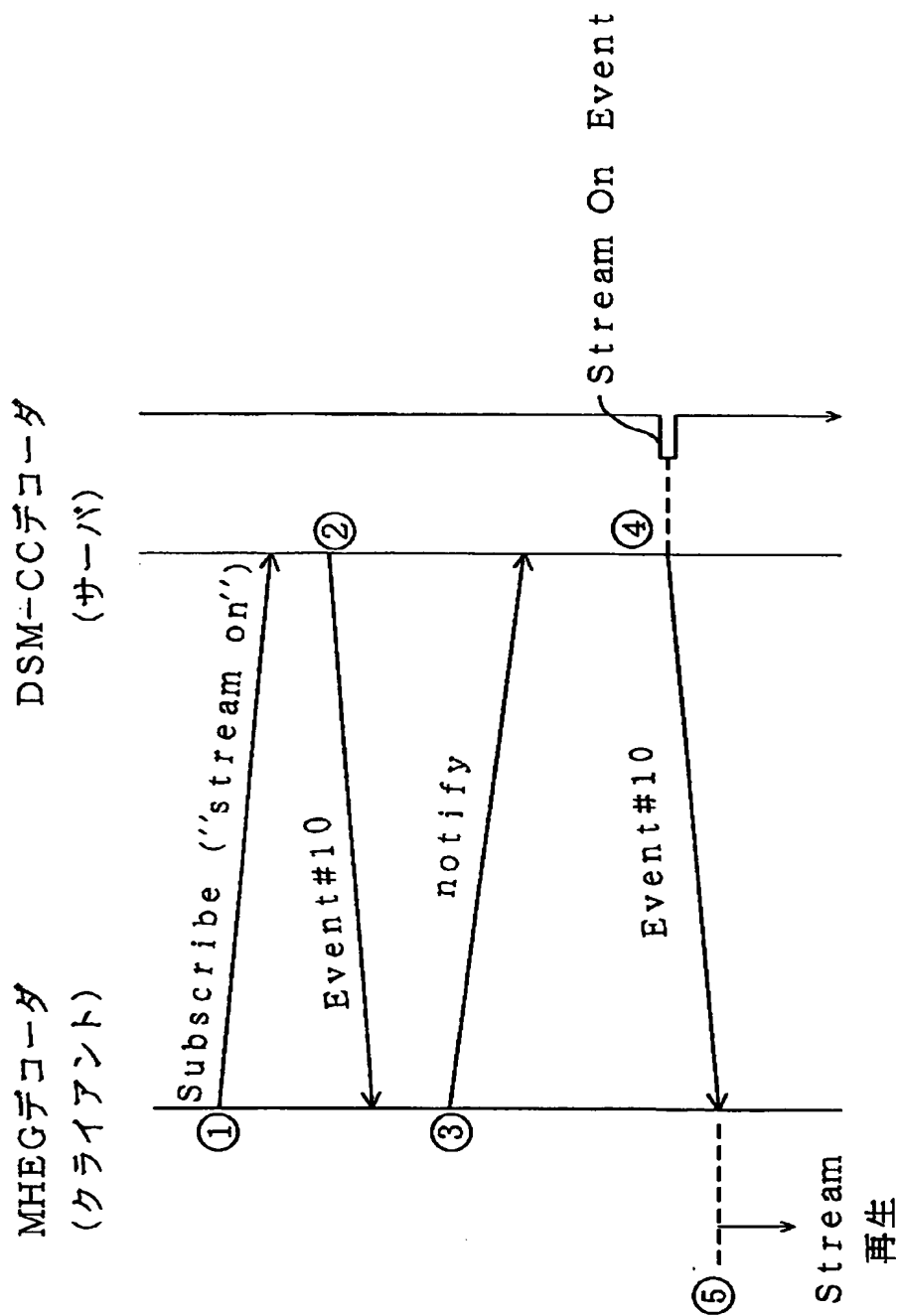
(b)

Update Event Id	module Id	module Ver. No

MHEGデコーダテーブル

DSM-CCデコーダテーブル

【図 18】



U-U API インターフェイス例

【書類名】 要約書

【要約】

【課題】 カルousel内のオブジェクトの更新が、サーバ側からクライアント（MHEGエンジン）側に通知されるようにする。

【解決手段】 クライアント側では、関心のあるオブジェクトIDを付加してモジュールの更新イベントの受け取りをサーバに宣言（subscribe）し、サーバでは、このsubscribeに応答して、固有のモジュール更新イベントのIDを設定してクライアントに返送するというインターフェイスを実行するようにされる。この後のクライアントのモジュール更新イベントの受け取り要求に応じて、サーバ側では、更新された制御情報のモジュール更新イベントIDを付加してモジュール変更イベントを送信するように構成する。クライアント側ではオブジェクトIDとモジュール更新イベントIDとを対応付けしたテーブルを検索することで、ほぼ更新されたオブジェクトの特定まで行うことが可能になる。また、クライアント側では獲得したモジュール更新イベントIDと対応するオブジェクトIDが一義的に得られるため、更新されたオブジェクトを特定するのにテーブルサーチを行う必要が無くクライアント側の処理負担は軽減される。

【選択図】 図12

【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000002185  
【住所又は居所】 東京都品川区北品川6丁目7番35号  
【氏名又は名称】 ソニー株式会社

【代理人】 申請人

【識別番号】 100086841  
【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル  
6階  
【氏名又は名称】 脇 篤夫

【代理人】

【識別番号】 100102635  
【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル  
6階 雄渾特許事務所  
【氏名又は名称】 浅見 保男

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号  
氏 名 ソニー株式会社